NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE KALMAN FILTER APPLIED TO PROCESS RANGE DATA OF THE CUBIC MODEL 40 AUTOTAPE SYSTEM

by

Benjamin E. Julian

December, 1976

Thesis Advisor: H. A. Titus

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

9

4. TITLE (and Subtitle)

6

The Kalman Filter Applied to Process Range Data of the Cubic Model 40 Autotape System.

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis

~~December, 1976~~

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

10

Benjamin E. Julian

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Montery California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

11

12. REPORT DATE

Dec[redacted]76

13. NUMBER OF PAGES

66

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

12 66p.

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Kalman Filter Autotape

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Kalman Filter is implemented to process range data output from the Cubic Model 40 Autotape system, a surface position locating system currently employed on the underwater tracking ranges at Dabob Bay and Nanoose. Results are presented for different measurement noise and forcing function noise statistics.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 450

The Kalman Filter Applied to Process Range Data of

the Cubic Model 40 Autotape System

by

Benjamin E. Julian
Lieutenant Commander, United States Navy
B.S., University of Washington, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1976

Author: Benjamin E Julian

Approved by: Harold A. Titus
Thesis Advisor

Donald E. Kirk
Chairman, Department of Electrical Engineering

Robert R. Fossum
Dean of Science and Engineering

ABSTRACT

The Kalman Filter is implemented to process range data output from the Cubic Model 40 Autotape system, a surface position locating system currently employed on the underwater tracking ranges at Dabob Bay and Nanoose. Results are presented for different measurement noise and forcing function noise statistics.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The Cubic CM-40 Autotape is a microwave distance measuring system used (by the U.S. Navy at its acoustic underwater tracking ranges at Dabob Bay and Nanoose) to provide reference position information for units on the surface and in the air above the range. This portable system consists basically of an interrogator which is operated aboard the unit to be tracked, two responders operated at two different shore sites and the associated antenna/RF assemblies. Required support systems include a data display and recording setup and an ADP facility for off-line processing of the Autotape data. Figure 1 shows the Autotape system components and Figure 2 shows a typical application geometry.

Historically, the Autotape has been used in such applications as tracking hydrophone array survey, buoy and hydrophone array planting and as a reference position indicator for calibrating other position-finding devices against. Generally, the Autotape has been used where an extremely high degree of accuracy is not required.

In operation, the system will provide for the display and recording of two ranges simultaneously, once per second, the ranges being those between the interrogator and each of the responders. The ranges are computed from the phase delay between the output of the modulation signal generator and a signal which has traveled from the interrogator to a responder and back. Ranging accuracy is stated by the manufacturer to be $\pm$ 0.5 meter + 10 ppm x range. Ranging frequencies of 1500 KHZ, 150 KHZ and 165 KHZ modulate a 3000 MHZ carrier, yielding a maximum unambiguous range of 10,000 meters with a resolution of 0.1 meter. However, independent

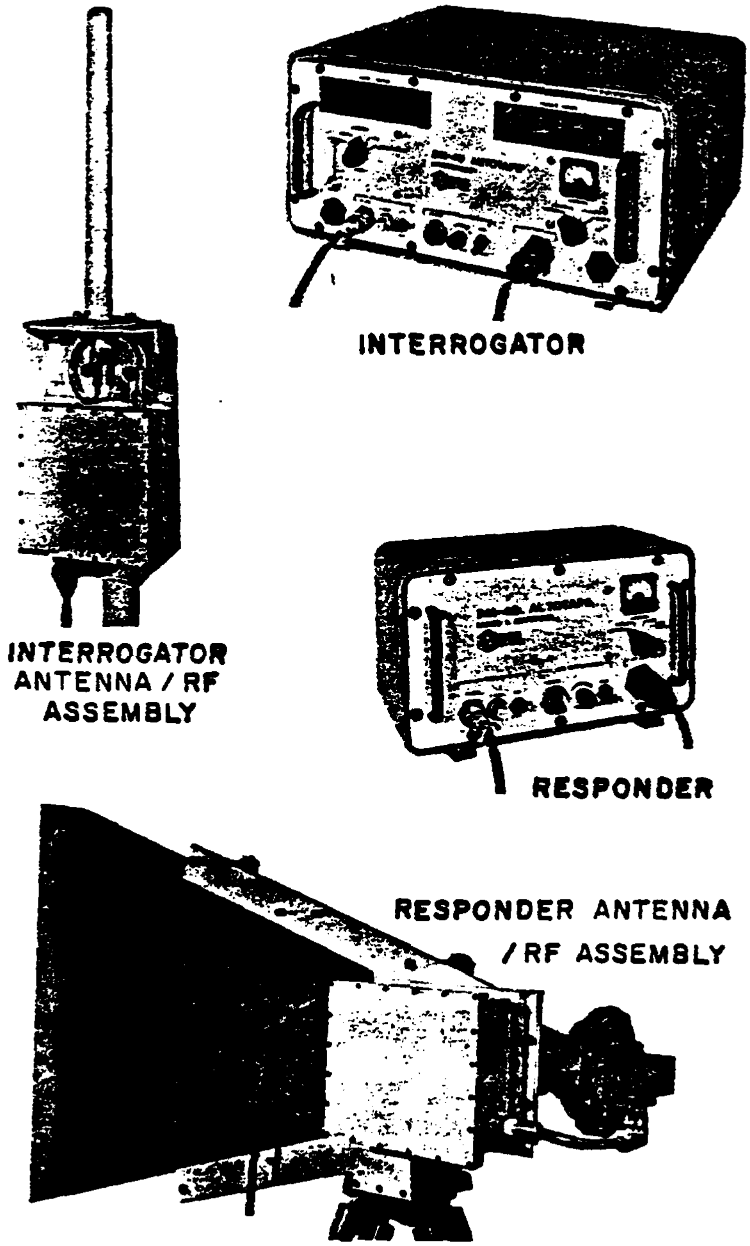


FIGURE 1: Cubic Model 40 Autotape System

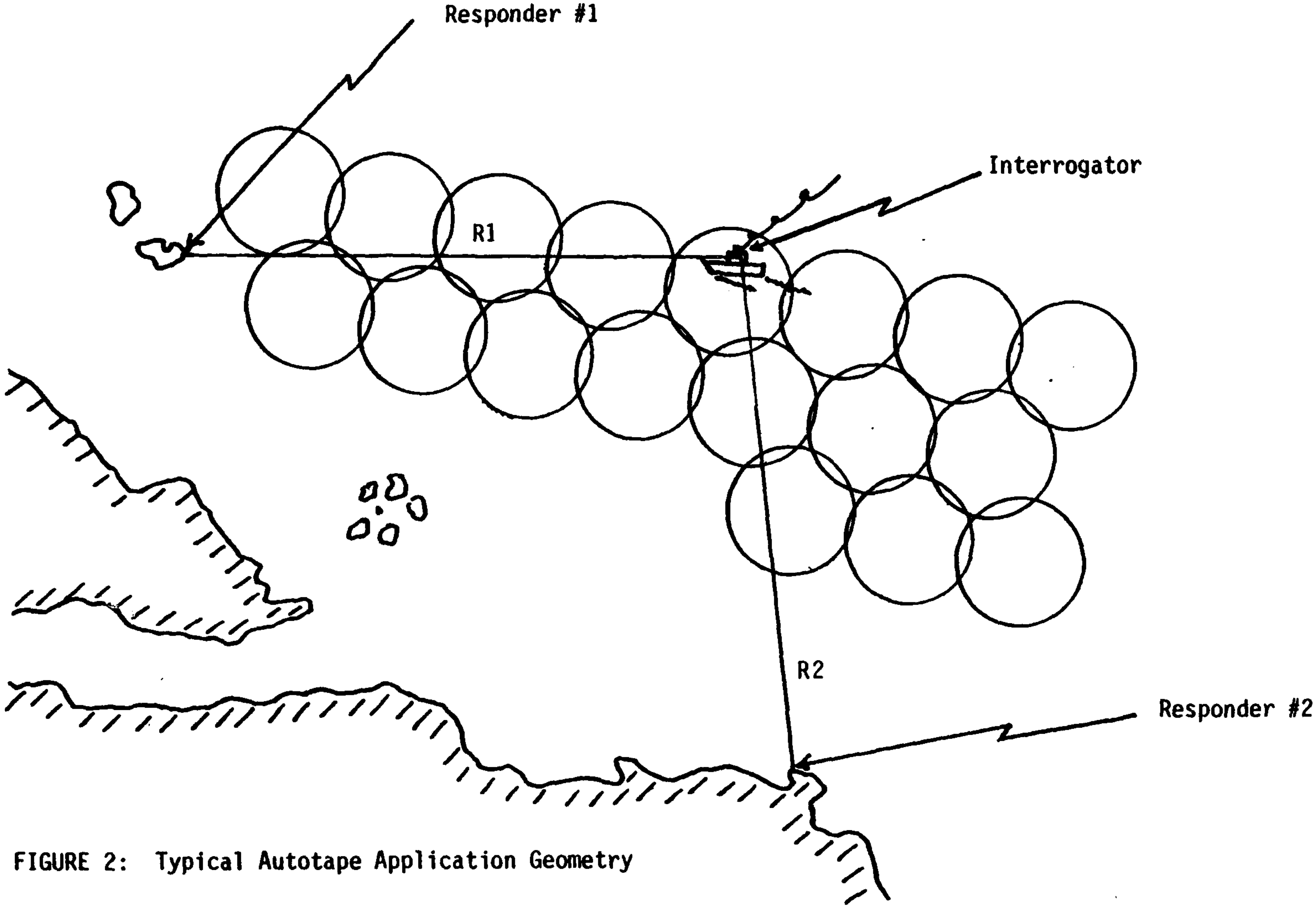


FIGURE 2: Typical Autotape Application Geometry

testing by the U.S. Navy [Reference 1] has shown that system accuracy may not be quite as good as stated by the manufacturer.

The accuracy of the Autotape system is principally dependent upon range errors, the geometry of the system and the method of data reduction. These factors are, in turn, affected by propagation velocity, system stability, range dependency, land survey accuracy, system geometry, slope reduction and data smoothing. A final anomaly which, depending upon the application, can substantially degrade the quality of the data-stream out is the orientation, over time, of the interrogator antenna in the vertical dimension. The interrogator antenna has only a 10 degree vertical beam width. Thus, if the system is being used on a platform such as a moderately maneuvering helicopter or a ship rolling substantially in the seaway, the system tends to frequently lose track, resulting in fairly long streams of useless data.

Present data reduction techniques employed when the system is used on either of the ranges (Dabob or Nanoose) employ two overall iterations. The first, or initial processing, administers the following three corrections to the raw range data:

1. Range Calibration Correction: This is a fixed value (meters) added to or subtracted from each range.
2. Propagation Velocity Correction: This is a variable correction due to the atmospheric index of refraction at the particular time and place of the exercise.
3. Slope Reduction Correction: This reduces both range measurements (which are actually slant ranges because the interrogator and the responders are not normally located at the exact same elevation) to a common horizontal plane at sea level.

Subsequent processing of the data includes conversion of the corrected ranges to a rectangular x-y range coordinate system and a moving average smoothing technique which employes curve fitting algorithms (linear,

parabolic or logarithmic) to reduce the data to its final form. Not uncommonly, as a result of the total reduction effort, the net remainder is an inadequate data package (in terms of quantity) for proper final evaluation.

Figure 3 is a rectangular plot of the raw ranges recorded during a recent array survey. The purpose of this project has been to design a filter, a Kalman filter, which would provide more accurate range data, as well as one that would track through the periods of "lost track" ranging, thereby providing a significantly larger final volume of data for evaluation. This paper presents the basic theory necessary and includes the final version of the filter.

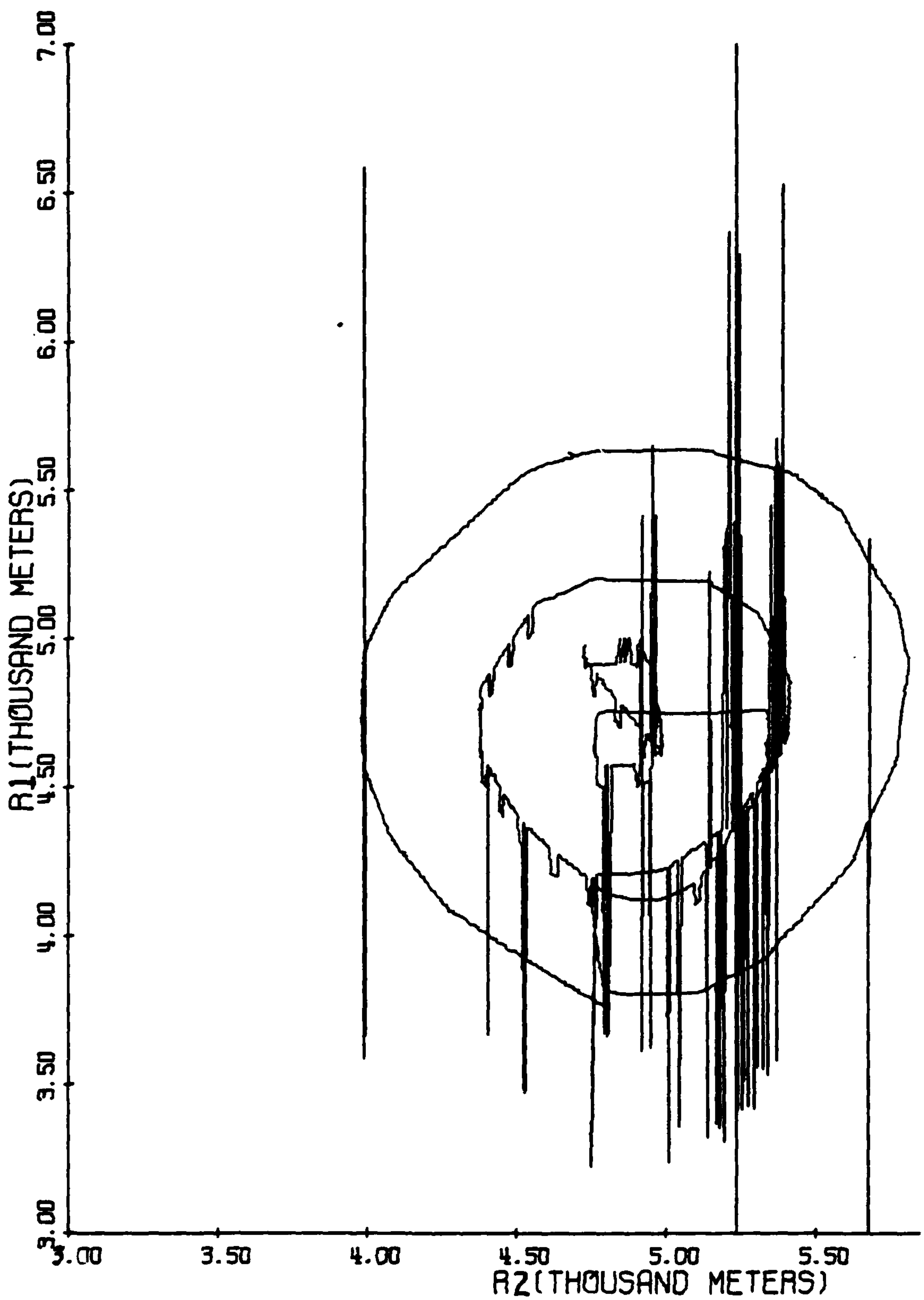


FIGURE 3: Rectangular Plot of Raw Range Data

II. THE FILTER THEORY AND DESIGN

A. THE SYSTEM DYNAMIC MODEL

A common application for the Autotape system is its use as a reference position locator on the surface unit conducting an acoustic hydrophone array (range) survey. The usual exercise plan will call for a service unit, carrying the interrogator and equipped with an acoustic pinger mounted on the underwater hull, to transit three concentric circular tracks, centered above the array, with track radii ranging from 100 to 1,000 meters, at speeds of up to eight knots. The direction of rotation for the outer track will normally be opposite to that of the middle circle. W!·le the service unit is being tracked via Autotape, it is also being tracked by the acoustic array. By comparing the acoustic position data with that from the Autotape, a digital computer is able to compute actual position and attitude of the array.

The desired estimates will be those of position and velocity, R1, R2, $\dot{R}1$, $\dot{R}2$. It is proper at this point to define a number of terms and to summarize some pertinent results of observer theory. First, we may define a fourth order state vector:

$$\underline{x} = \begin{bmatrix} R1 \\ R2 \\ \dot{R}1 \\ \dot{R}2 \end{bmatrix}$$

Recall that a linear system can be described in the continuous time domain as:

$$\dot{\underline{x}}\,(t) = \underline{A}\;\underline{x}\,(t) + \underline{D}\;\underline{w}\,(t)$$

where: $\underline{x}(t)$ is the n-element column vector of the states

$\underline{A}$ and $\underline{D}$ are nxn and nxp matrices describing system dynamics

$\underline{w}(t)$ is a q-element vector of random noise inputs to the system

The system measurements may be expressed as:

$$\underline{z}(t) = \underline{H}\ \underline{x}\ (t) + \underline{v}\ (t)$$

where: $\underline{z}(t)$ is the q-element vector of system measurements

$\underline{H}$ is the qxn weighting matrix for the measurements

$\underline{v}(t)$ is the q-element vector of random measurement noise

The corresponding linear discrete model may be written as:

$$\underline{x}(k+1) = \underline{\phi}\ \underline{x}(k) + \underline{\Gamma}\ \underline{w}(k)$$

with no deterministic inputs to the system.

Also, $\underline{z}(k) = \underline{H}\ \underline{x}(k) + \underline{v}(k)$

For the system under consideration, it can be shown that the state transition matrix

$$\underline{\phi} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $$\underline{\Gamma} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

for a sampling interval T of 1 second. A block diagram of the system is shown in Figure 4.

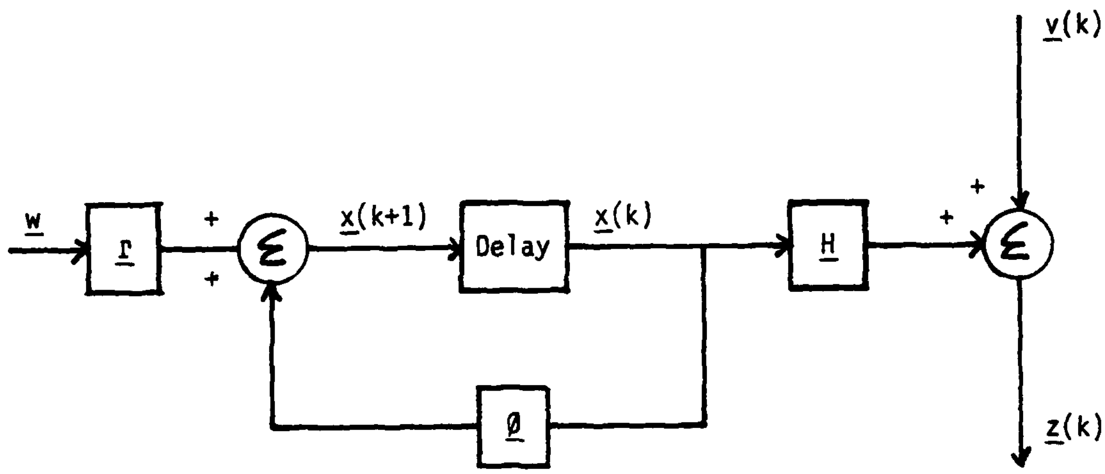


FIGURE 4: Block Diagram of Discrete Linear Estimator

The following assumptions will be made regarding the noise processes and the initial state, $\underline{x}(o)$ of the plant [Ref. 2]:

The measurement noise has zero mean, is uncorrelated, and

$$E[\underline{v}(k)\ \underline{v}^T(j)] = \underline{R}(k)\ \delta_{kj},$$ where δ is the kronecker delta

The forcing noise has zero mean, is uncorrelated, and

$$E[\underline{w}(k)\ \underline{w}^T(j)] = \underline{Q}(k)\ \delta_{kj}$$

The forcing noise and measurement noise are uncorrelated.

The initial state is a random variable with known mean and covariance, and

$$E[\{\underline{x}(o) - \bar{\underline{x}}_o\}\ \{\underline{x}(o) - \bar{\underline{x}}_o\}^T] = \underline{P}_o$$

The measurement noise and initial state are uncorrelated.

The forcing noise and initial state are uncorrelated.

The Kalman Filter equations and their derivation are well known [Ref. 2], [Ref. 3]:

$$\underline{G}(k) = \underline{P}(k/k-1)\ \underline{H}^T(k)\ [\underline{H}(k)\ \underline{P}(k/k-1)\ \underline{H}^T(k) + \underline{R}(k)]^{-1} \quad (1)$$

$$\underline{P}(k/k-1) = \underline{\phi}\ \underline{P}(k-1/k-1)\ \underline{\phi}^T + \underline{Q} \quad (2)$$

$$\underline{P}(k/k) = [\underline{I} - \underline{G}(k)\ \underline{H}(k)]\ \underline{P}(k/k-1) \quad (3)$$

$$\hat{\underline{x}}(k/k) = \underline{x}(k/k-1) + \underline{G}(k)\ [\underline{z}(k) - \underline{H}(k)\ \underline{x}(k/k-1)] \quad (4)$$

$$\hat{\underline{x}}(k/k-1) = \underline{\phi}(k/k-1)\ \underline{x}(k-1/k-1) + \underline{\Gamma}(k/k-1)\ \underline{w}(k-1) \quad (5)$$

Where the notation (k/k-1) interprets as the value of the parameter of note <u>at time k</u> given measurements at times up to and including time k-1. (k/k) and (k-1/k-1) have similar interpretations. The $\hat{\underline{x}}$ denotes the estimate of $\underline{x}$.

$\underline{G}(k)$ represents the filter gain at time k. $\underline{P}$ represents the covariance of estimation error;

$$\underline{P}\,(k/k) = E\,[\underline{e}(k/k)\,\underline{e}^T(k/k)] = E\left\{\begin{bmatrix} e_1(k/k) \\ e_2(k/k) \\ \vdots \\ e_n(k/k) \end{bmatrix} [e_1(k/k)\ e_2(k/k) --- e_n(k/k)]\right\}$$

$$= E\left\{\begin{bmatrix} e_1^2(k/k) & e_1(k/k)\,e_2(k/k) & --- & e_1(k/k)\,e_n(k/k) \\ e_2(k/k)\,e_1(k/k) & e_2^2(k/k) & --- & e_2(k/k)\,e_n(k/k) \\ \vdots & & & \\ e_n(k/k)\,e_1(k/k) & e_n(k/k)\,e_2(k/k) & --- & e_n^2(k/k) \end{bmatrix}\right\}$$

where $\underline{e}(k/k) = \underline{\hat{x}}(k/k) - \underline{x}(k)$. A complete standard block diagram for the filter and an information flow diagram are included as Figures 5 and 6 as slightly different viewpoints from which the system may be viewed and understood. Figure 7 shows a timing diagram of the various quantities contained in the filter equations.

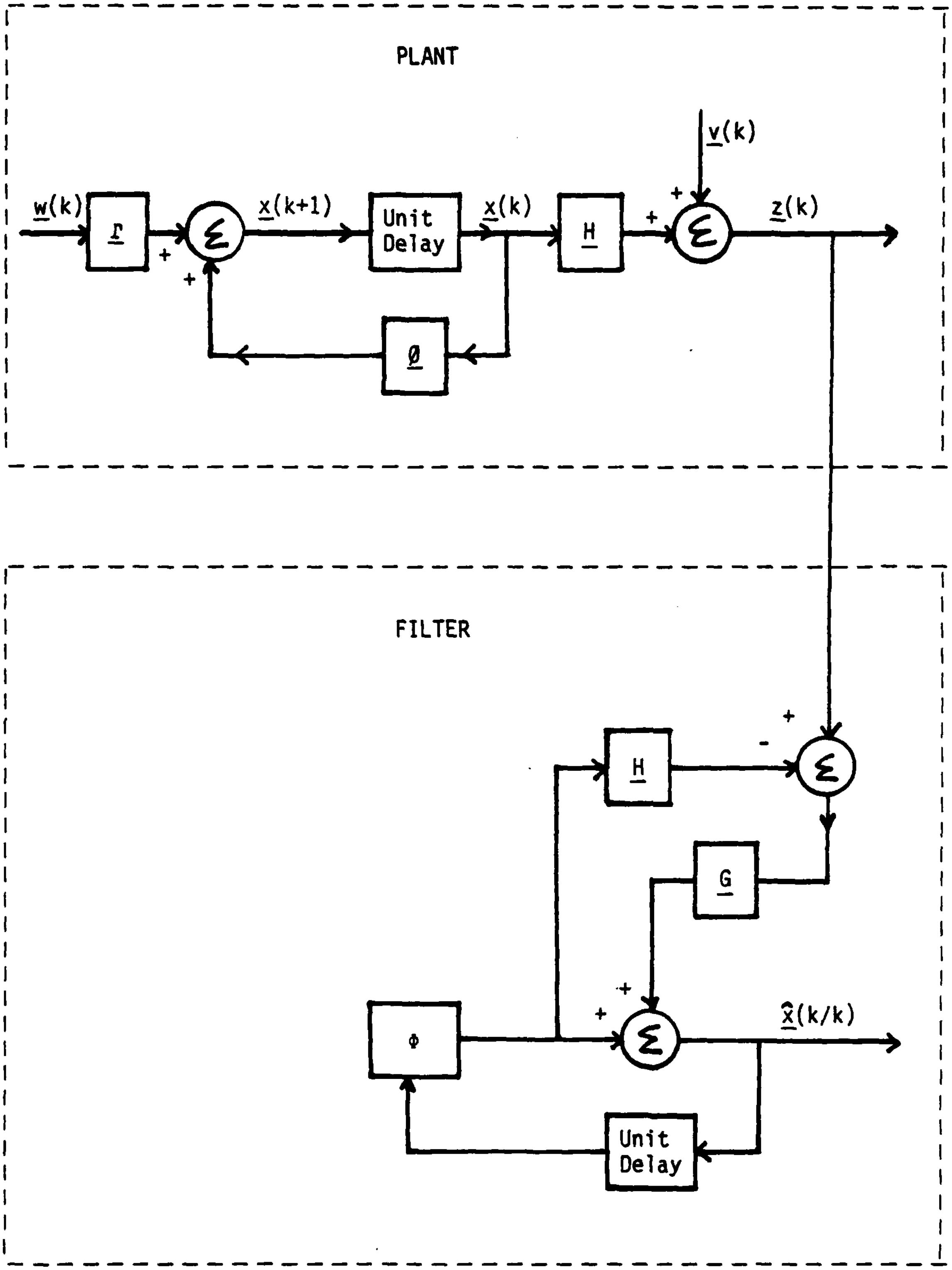


FIGURE 5: Kalman Filter Block Diagram

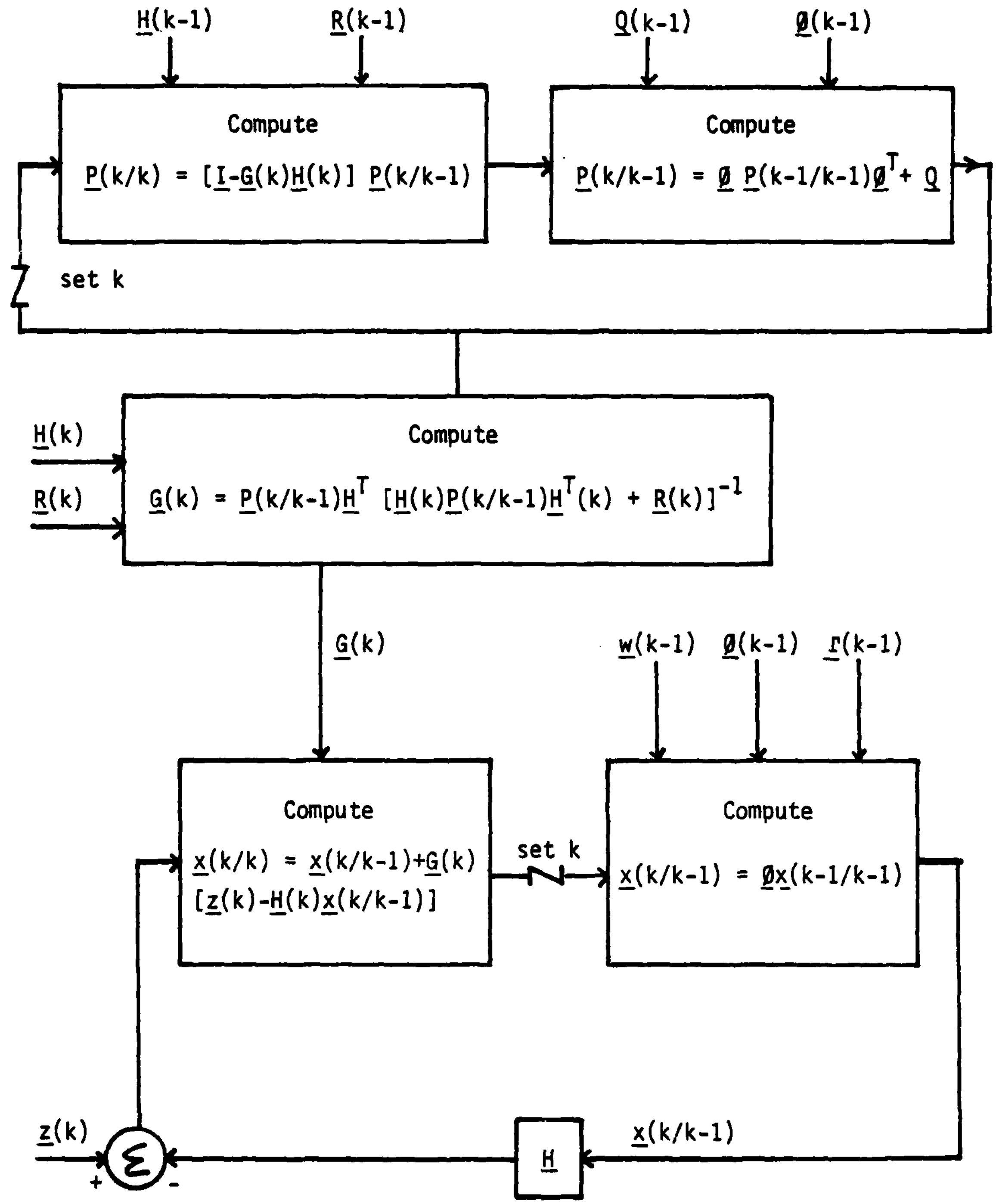


FIGURE 6: Simplified Information Flow Diagram of a Discrete Kalman Filter

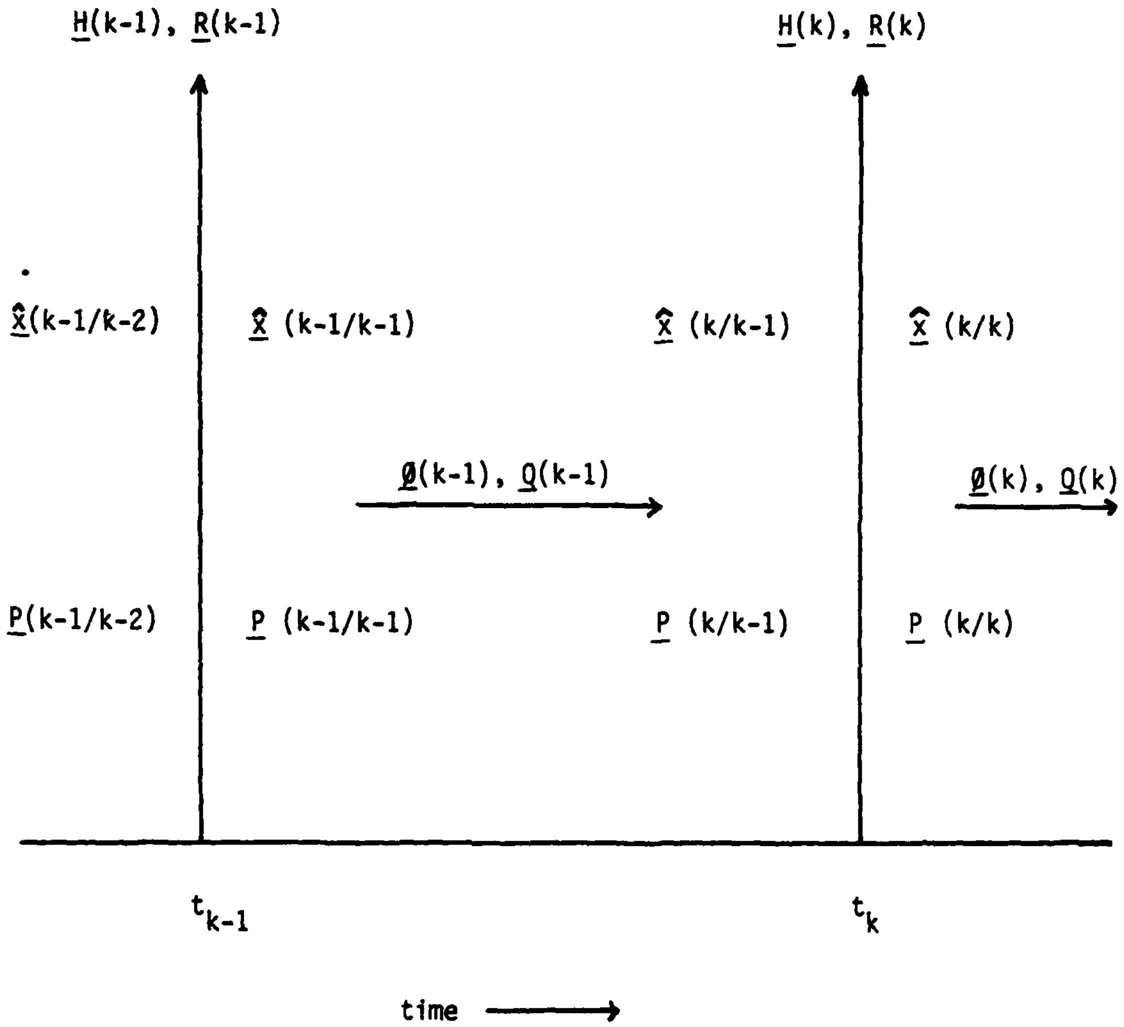


FIGURE 7: Timing Diagram of Filter Equation Quantities

B. THE PROCESSOR

Appendix A is a flowchart of the Kalman filter program utilized. Initially, the matrices describing the physical system, the noise statistics and other program parameters are read into storage and printed out. The discrete state-transition matrix, Phi, is computed and printed out and the gain schedule is computed and printed out. It is seen that the elements of the gain matrix reach a steady state, and, for example, with both the $\underline{R}$ and $\underline{Q}$ matrices being identity matrices, the gain reaches steady state between k=5 and k=10. Therefore, in the main iteration loop, the filter will essentially be a constant gain filter for $k > 10$.

Next, the main iteration loop commences. The initial measurements are read and x1(0/-1) and x2(0/-1) are initialized to these values. x3(0/-1) and x4(0/-1), representing the rates, are set to the mean constant value (in the respective directions) of 4.0 meters per second. The Autotape output is a 5 significant figure output, modulo 10,000, reading to 0.1 meter. Inherent in the output is a major degree of jitter in the two most significant digits, which would significantly distort the covariance of measurement noise. Therefore, as an option, measurements could be gated, and the gain automatically set to zero in those cases where the residue falls outside of a maximum reasonable bound.

Commencing with k=0, and utilizing the known values for $\underline{\hat{x}}(0/-1)$ and $\underline{P}(0/-1)$, the Kalman filter equations are solved iteratively in the following manner [see page 15, equations (1)-(5)]:

(1), (3), (4),
Increment k to k=1
(5), (2), (1), (3), (4),
Increment k to k=2
(5), (2), (1), (3), (4),
etc.

Also computed on each iteration are the error residues:

$$\underline{RES} = \underline{z} - \underline{x}(k/k-1)$$

and the one-step prediction errors:

$$\underline{ERR} = \underline{x}(k/k) - \underline{x}(k/k-1)$$

Finally, the computations are tabulated and plots are produced.

C. NOISE AND ERROR CONSIDERATIONS

Reference 1 documents an Autotape evaluation which was conducted in 1971. The error geometry is shown in Figure 8. Graphically, position is determined by locating the crossing point of the two range arcs, in conjunction with a knowledge of the baseline formed by the two responders. Since each range has an associated standard deviation (error), the point can actually be enclosed in a parallelogram which defines the probable position within one standard deviation of the ranges. The shape of the parallelogram will vary with the position of the crossing point relative to the baseline, as indicated in Figure 8. It can be shown that the maximum probable error (MPE) will be minimized where the range arcs are orthogonal. Figure 9 diagrams error contours which are actually the locii of constant MPE for two particular responder sites on the Nanoose Range. Table 1 summarizes pertinent results of the study.

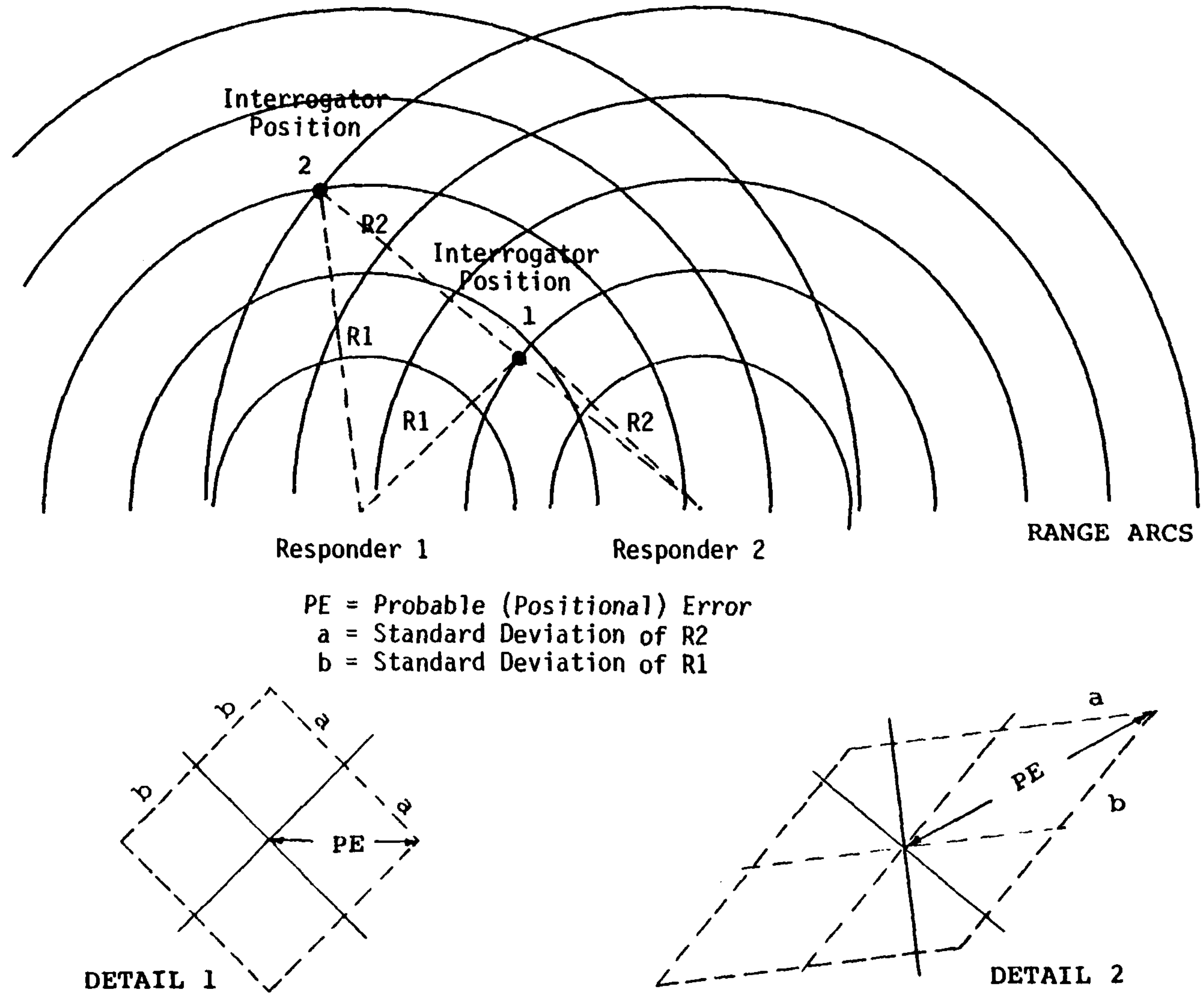


FIGURE 8: Error and Geometry. At interrogator position 1, the range arcs are nearly orthogonal, and MPE is minimized. At interrogator position 2, the range arcs are not orthogonal, and MPE is greater.

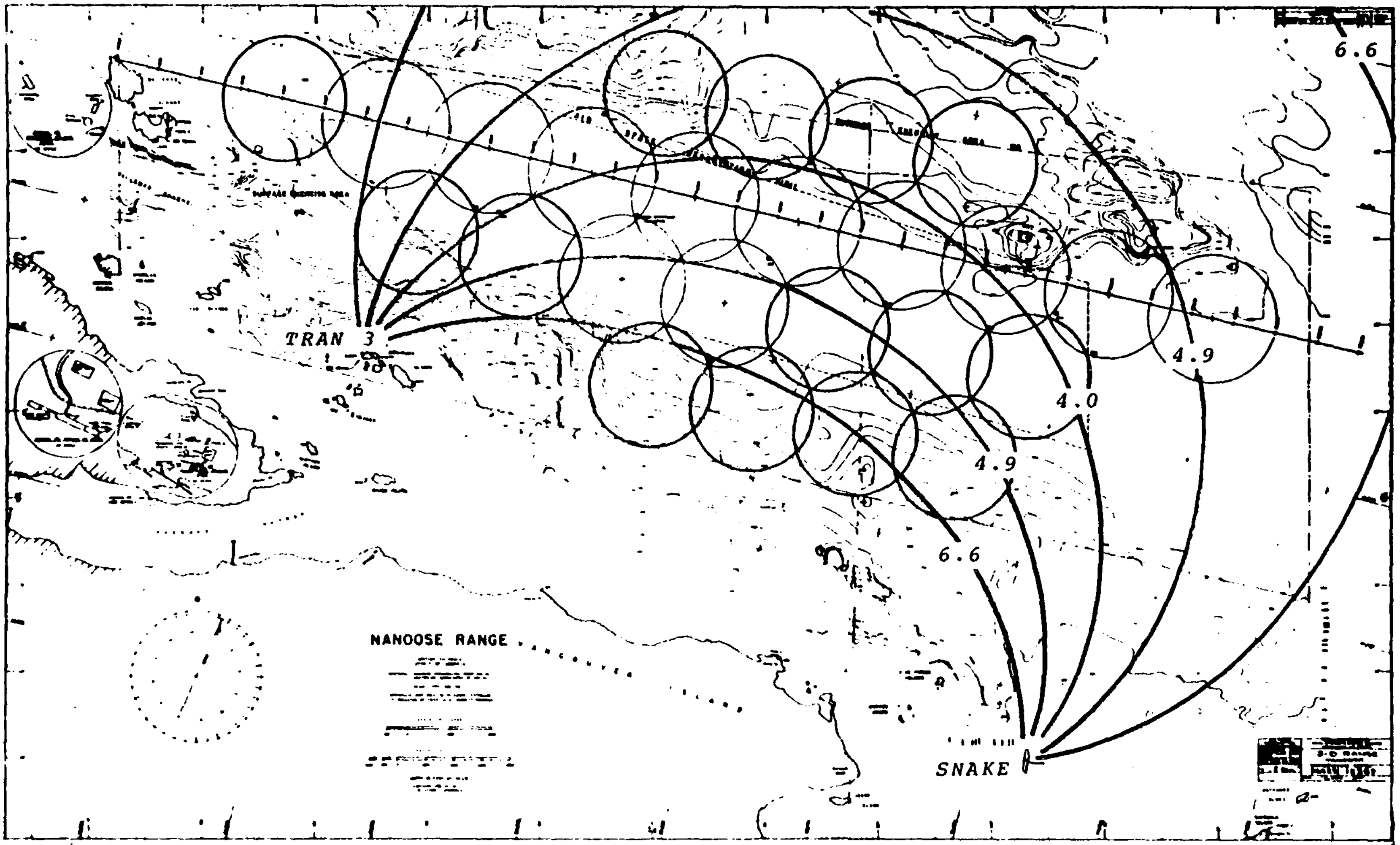


FIGURE 9: Error Contours (Arcs represent maximum probable positional error in feet. End points of arcs are responder locations.)

TABLE 1					
Average Range Errors (feet)					
		R-1		R-2	
Survey	No. Points	Error Average	Standard Deviation	Error Average	Standard Deviation
Array 04	30	- 0.5	2.8	- 0.1	2.8
Array 07	49	- 1.3	2.3	- 0.4	2.4
Array 08	10	- 0.8	4.4	1.6	2.8
Array 09	25	3.8	2.6	0.	2.2
Average		0.3	3.0	- 0.5	2.6

For the purpose of modeling the covariance of excitation noise, it was assumed that the service unit transited an 800 meter circle at an average speed of eight knots. Then:

$$a = \frac{v^2}{R} = \frac{\left[\dfrac{(8\text{ kts})\left(\dfrac{1830\text{ meter}}{\text{n. mile}}\right)}{3600\ \dfrac{\text{sec}}{\text{Hr}}}\right]^2}{800\text{ meters}}$$

$$= .0207\ \frac{\text{m}}{\text{sec}^2}$$

Filter performance was investigated for $\underline{Q} = \underline{I}$, $.1\underline{I}$, and $.01\underline{I}$, for

$$\underline{P}(0/-1) = \underline{P}_0 = E\left\{\left[\underline{x}(0) - \bar{\underline{x}}_0\right]\left[\underline{x}(0) - \bar{\underline{x}}_0\right]\right\} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $$\underline{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

where the a priori $\underline{x}(0/-1)$ is known to be a reasonably good estimate -- approximately the same accuracy as an observation.

D. PROCESSOR PERFORMANCE; AUTHOR'S CONCLUSIONS

Table 2 summarizes a comparison of the Kalman filter performance with the results of the (corrected) processing by the program presently being used for the cases $\underline{Q} = \underline{I}$, $\underline{R} = \underline{I}$, and $\underline{Q} = 0.1\underline{I}$, $\underline{R} = \underline{I}$. Figures 10, 11, 12 and 13 are residue and error plots for the example $\underline{Q} = .01\underline{I}$, $\underline{R} = \underline{I}$.

It is seen that the Kalman filter will satisfactorily handle the data where the measurement noise statistics approximate those used in the model. However, for the noise resulting from the jitter which appears in the "hundreds" and "thousands" digits, the filter, as configured without a gate, will estimate with considerable error. The raw range

R2 was clean of this particular noise element, and the results as indicated by Figures 12 and 13 were superior to those for R1.

It is suggested that the Kalman filter be used as the first iteration processing of the Autotape output.

TABLE 2

TABULATED PROCESSOR COMPARISON

TIME	RAW		CURRENT PROCESSOR		KALMAN FILTER			
			Smoothed		Q=1.0		Q=0.1	
	R1	R2	R1	R2	R1	R2	R1	R2
105543	4639.9	4962.2	4640.95	4964.94	4640.5	4962.9	4640.1	4965.0
105733	4911.2	4804.4	4911.72	4806.86	4911.6	4805.1	4912.0	4805.7
105828	4860.7	4967.3	4858.71	4966.50	4862.3	4967.5	4859.2	4967.5
105855	4732.6	4982.2	4730.33	4981.69	4732.2	4982.2	4732.5	4982.1
105915	4628.9	4984.0	4630.19	4986.86	4629.4	4984.7	4630.3	4986.2
105950	4572.6	4846.7	4571.74	4844.70	4572.9	4846.5	4573.5	4846.1
110023	4656.7	4766.5	4652.96	4763.53	4656.1	4766.4	4654.6	4765.8
110042	4741.2	4774.9	4738.52	4773.05	4740.9	4774.3	4741.2	4773.0
110057	4755.3	4822.8	4754.26	4822.27	4755.1	4822.4	4755.6	4822.1
110109	4750.0	4872.8	4748.80	4872.46	4749.7	4872.2	4749.5	4871.7
110116	4748.1	4904.3	4747.17	4903.95	4748.1	4904.2	4748.1	4904.0
110146	4748.6	5050.7	4744.84	5047.81	4748.0	5050.2	4747.6	5049.3
110332	3550.6	5326.3	4550.12	5325.59	3729.8	5326.1	5326.1	3979.7
110501	4165.0	5079.2	4164.08	5079.49	4167.0	5079.6	4174.4	5080.3
110825	4720.0	4378.9	4718.36	4378.39	4721.5	4378.6	4728.0	4378.8
111001	5122.1	4573.4	5121.11	4574.53	5122.5	4573.7	5127.9	4573.7
111101	5196.0	4842.1	5193.48	4840.03	5195.9	4841.9	5195.6	4841.4
111315	4985.2	5352.0	4983.58	5352.20	4984.8	5351.6	4984.6	5351.8
111554	4306.8	5121.5	4303.26	5119.96	4317.9	5121.3	4288.9	5120.9
111632	4216.3	4954.0	4212.63	4952.43	4215.7	4953.9	4215.9	4953.4
112224	4406.6	5685.2	4357.80	5664.59	4471.0	5685.2	4359.7	5664.5
112343	4733.0	5786.8	4730.69	5784.93	4732.8	5786.2	4732.4	5785.0
112642	5552.4	5457.2	5455.20	5550.70	5457.1	5552.4	5457.0	5552.1
112758	5255.5	5602.2	5600.75	5255.58	5602.1	5255.5	5602.1	5255.5
112938	4766.1	5632.4	5630.61	4765.51	5632.5	4766.0	5632.6	4765.9
113121	5347.8	4294.9	5347.12	4295.95	5347.9	4294.9	5348.0	4295.2
113332	4789.6	3985.9	4788.31	3985.21	4789.5	3985.7	4789.7	3985.7
113511	4297.8	4101.5	4296.09	4102.81	4298.0	4102.3	4298.0	4102.8

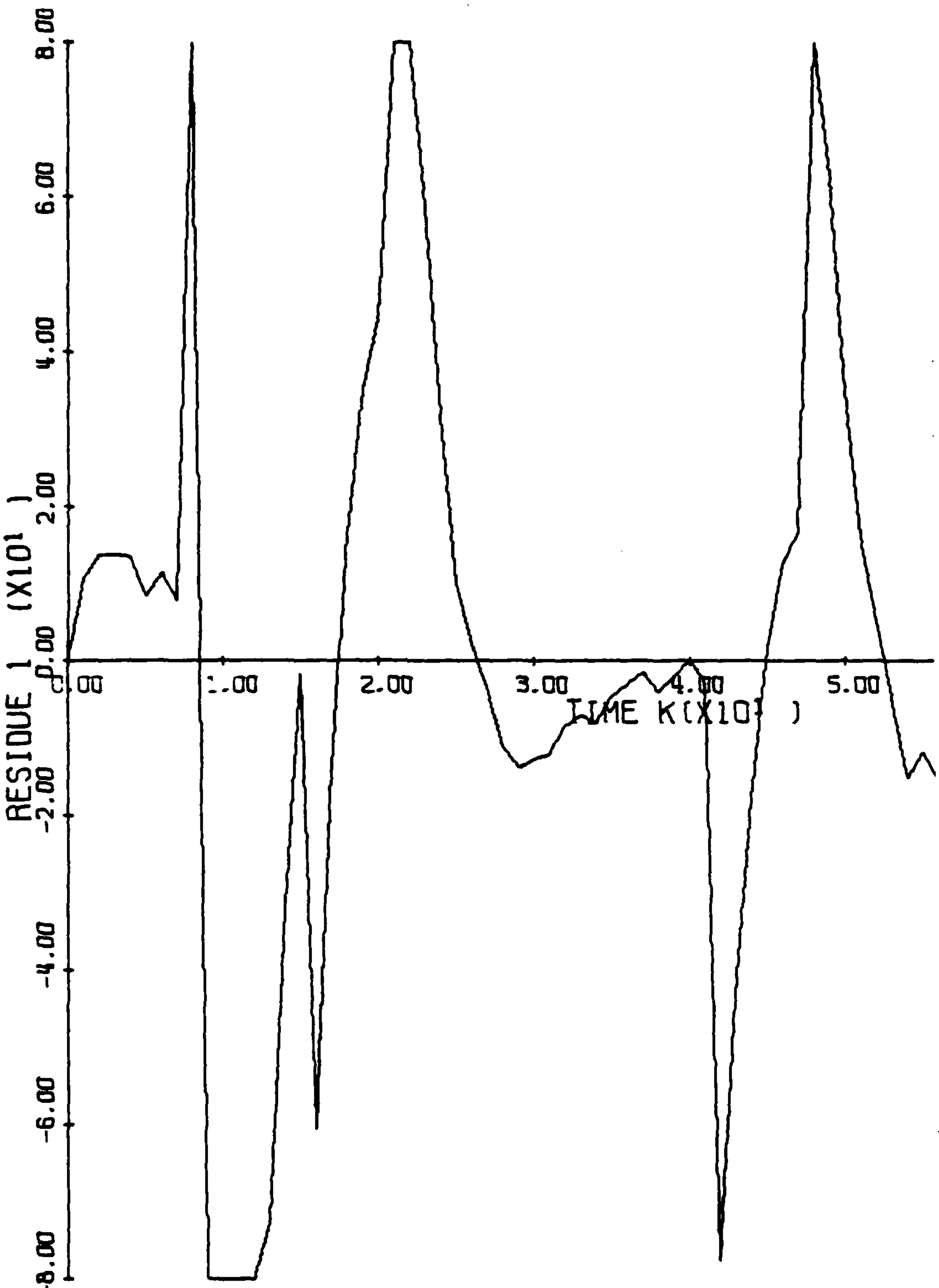


FIGURE 10: Residue 1 vs. Time. $\underline{Q} = .01\underline{I}$, $\underline{R} = \underline{I}$.

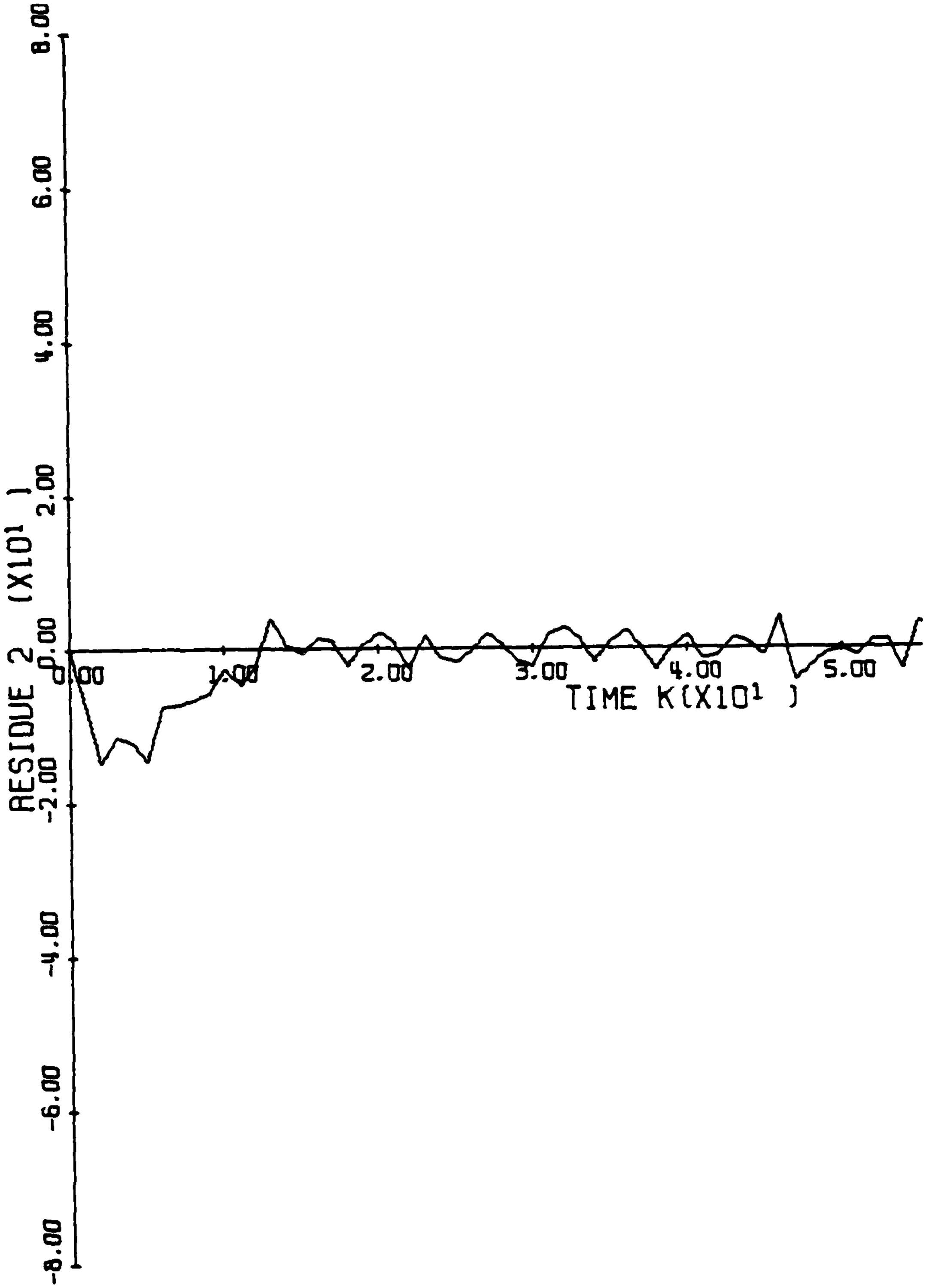


FIGURE 11: Residue 2 vs. Time. $\underline{Q} = .01\underline{I}$, $\underline{R} = \underline{I}$.

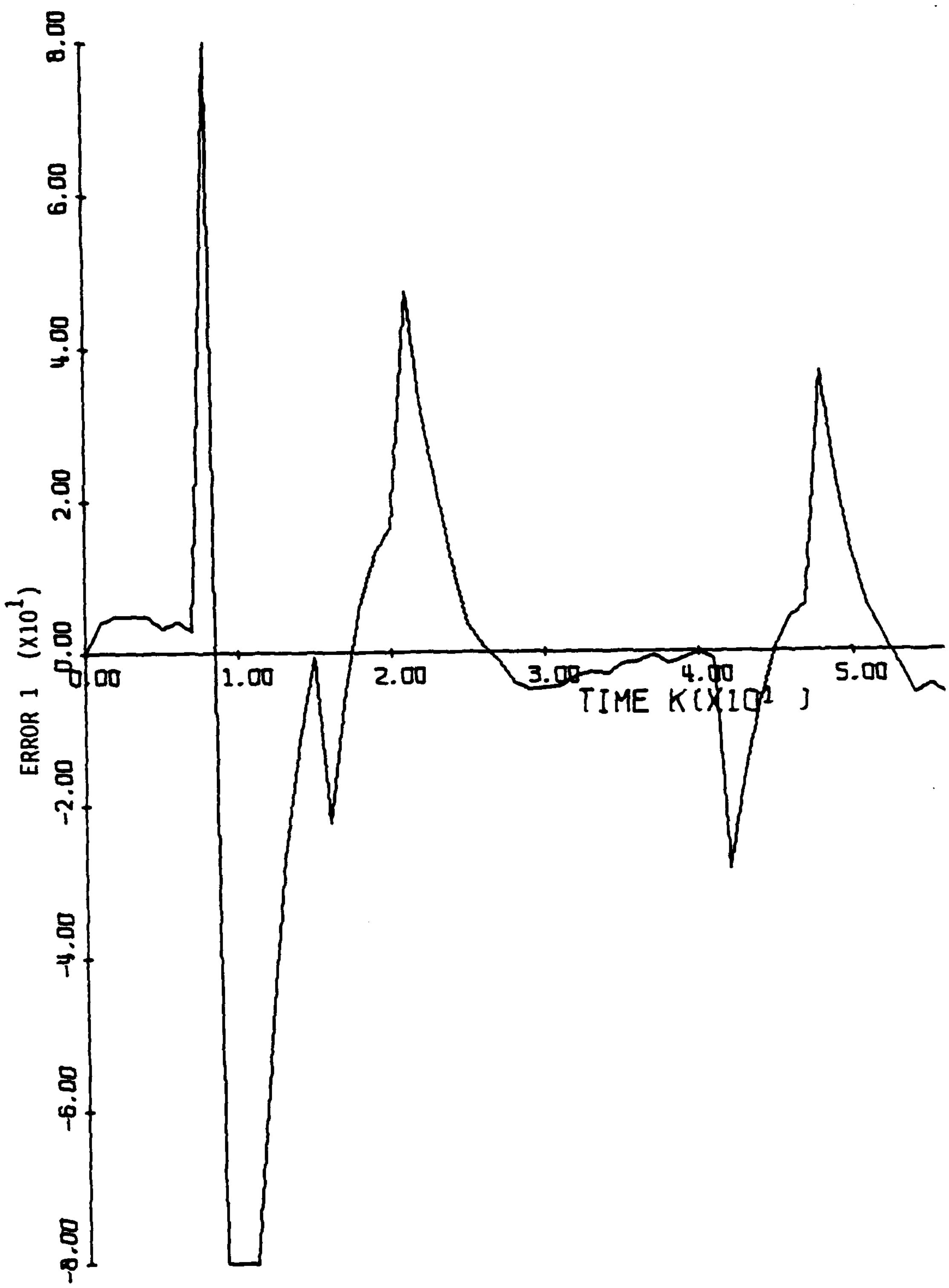


FIGURE 12: Error 1 vs. Time. $\underline{Q} = .01\underline{I}$, $\underline{R} = \underline{I}$.

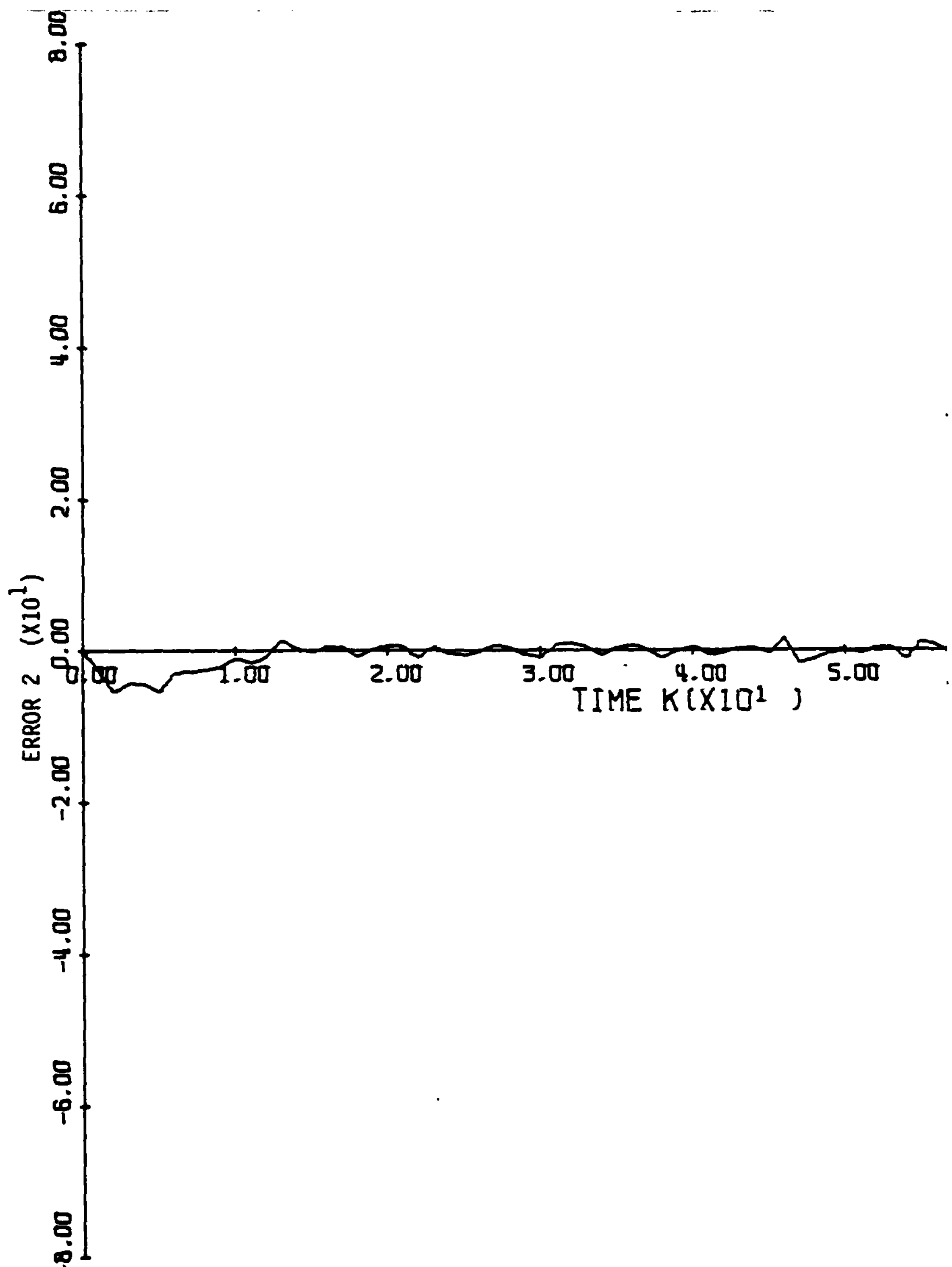


FIGURE 13: Error 2 vs. Time. $\underline{Q} = .01\underline{I}$, $\underline{R} = \underline{I}$.

III. FUTURE FILTER IMPROVEMENTS

The filter, as designed, will process by off-line (forward) filtering of the range measurements. It is suggested that, as an effort to further improve upon the quality of the processed data, a fixed-interval smoothing algorithm (the initial and final times, 0 and T, are fixed, and the estimate $\hat{\underline{x}}$ (t/T) is sought) be incorporated.

For the system and measurements described by:

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}\underline{w}$$

$$\underline{z} = \underline{H}\underline{x} + \underline{v}$$

the equations defining the forward filter are, in the time domain [Ref.3]:

$$\dot{\hat{\underline{x}}} = \underline{F}\hat{\underline{x}} + \underline{P}\underline{H}^T\underline{R}^{-1}\,[\underline{z} - \underline{H}\underline{x}], \quad \hat{\underline{x}} = \hat{\underline{x}}_o \qquad (1)$$

$$\dot{\underline{P}} = \underline{F}\underline{P} + \underline{P}\underline{F}^T + \underline{G}\underline{Q}\underline{G}^T - \underline{P}\underline{H}^T\underline{R}^{-1}\underline{H}\underline{P}, \quad \underline{P}(0) = \underline{P}_o \qquad (2)$$

To write the backward filter equations, set $\tau = T - t$. Then $\frac{dx}{d\tau} = \frac{-dx}{dt}$, and

$$\frac{dx}{d\tau} = -\underline{F}\underline{x} - \underline{G}\underline{w}, \quad \text{for } 0 \leq \tau \leq T,$$ denoting differentiation with respect to backward time.

Also,

$$\underline{z}(\tau) = \underline{H}\underline{x} + \underline{v}.$$

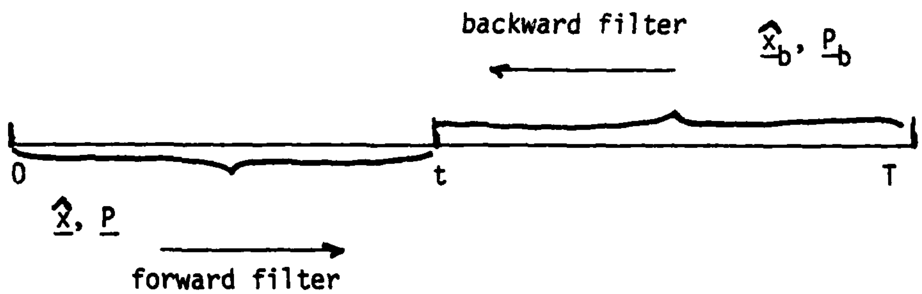
Then, by analogy, the backward filter equations can be written by changing $\underline{F}$ to $-\underline{F}$ and $\underline{G}$ to $-\underline{G}$, resulting in:

$$\frac{d}{d\tau}\hat{\underline{x}}_b = -\underline{F}\hat{\underline{x}}_b + \underline{P}_b\underline{H}^T\underline{R}^{-1}\,[\underline{z} - \underline{H}\hat{\underline{x}}_b]$$

and

$$\frac{d}{d\tau}\underline{P}_b = -\underline{F}\underline{P}_b - \underline{P}_b\underline{F}^T + \underline{G}\underline{Q}\underline{G}^T - \underline{P}_b\underline{H}^T\underline{R}^{-1}\underline{H}\underline{P}_b \qquad (3)$$

FIGURE 14: Relationship of Forward and Backward Filters



From Figure 14, it can be seen that the smoothed estimate at time=T must be the same as the forward filter estimate at that point, i.e.,

$$\hat{\underline{x}}\ (T/T) = \hat{\underline{x}}\ (T)$$

and $$\underline{P}\ (T/T) = \underline{P}\ (T)$$

which yields the required boundary condition on $\underline{P}_b^{-1}$,

$$\underline{P}_b^{-1}\ (t=T) = \underline{0}, \quad \text{or } \underline{P}_b^{-1}\ (\tau=0) = \underline{0} \tag{4}$$

with the boundary condition on $\hat{\underline{x}}_b$ (T) not yet known. Therefore, define the new variable:

$$\underline{s}(t) = \underline{P}_b^{-1}\ (t)\ \hat{\underline{x}}_b\ (t) \tag{5}$$

and since $\hat{\underline{x}}_b$ (T) is finite, it follows that:

$$\underline{s}\ (t=T) = \underline{0}, \quad \text{or } \underline{s}\ (\tau=0) = \underline{0}. \tag{6}$$

Reformulation in terms of $\underline{P}_b^{-1}$ yields:

$$\frac{d}{d\tau} \underline{P}_b^{-1} = -\underline{P}_b^{-1} \left(\frac{d}{d\tau} \underline{P}_b \right) \underline{P}_b^{-1}$$

Thus, equation (3) can be written as:

$$\frac{d}{d\tau} \underline{P}_b^{-1} = \underline{P}_b^{-1} \underline{F} + \underline{F}^T \underline{P}_b^{-1} - \underline{P}_b^{-1} \underline{G}\, \underline{Q}\, \underline{G}^T \underline{P}_b^{-1} + \underline{H}^T \underline{R}^{-1} \underline{H} \qquad (7)$$

for which equation (4) is the appropriate boundary condition.

Differentiating equation (5) with respect to τ, and with some substitution and manipulation, we arrive at:

$$\frac{d}{d\tau} \underline{s} = \left(\underline{F}^T - \underline{P}_b^{-1} \underline{G}\, \underline{Q}\, \underline{G}^T \right) \underline{s} + \underline{H}^T \underline{R}^{-1} \underline{z} \qquad (8)$$

for which equation (6) is the appropriate boundary condition. Equations (1), (2), (7) and (8), along with:

$$\underline{P}^{-1} (t/T) = \underline{P}^{-1} (t) + \underline{P}_b^{-1} (t)$$

$$\underline{x} (t/T) = \underline{P} (t/T) [\underline{P}^{-1} (t)\, \hat{\underline{x}} (t) + \underline{P}_b^{-1} (t)\, \hat{\underline{x}}_b (t)]$$

define the optimal smoother.

Many forms of the smoothing equations may be derived. The form proposed for use in this particular case is the Rauch-Tung-Striebel form, with the discrete-time expressions summarized as follows:

Smoothed State Estimate $\hat{\underline{x}}(k/N) = \hat{\underline{x}}(k/k) + \underline{A}_k\, [\hat{\underline{x}}(k+1/N) - \hat{\underline{x}}(k+1/k)]$

where

$$\underline{A}_k = \underline{P}(k/k)\, \underline{\phi}(k)^T\, \underline{P}(k+1/k)^{-1}$$

for $k = N-1$

Error Covariance Matrix Propagation $\underline{P}(k/N) = \underline{P}(k/k) + \underline{A}_k\, [\underline{P}(k+1/N) - \underline{P}(k+1/k)]\, \underline{A}_k^T$

also for $k = N-1$

Solution of the equations would proceed as follows: As an example, and because it is slightly easier to see when actual times are used, suppose NN = 100. On the forward filter pass, the values of $\hat{\underline{x}}(k/k)$, $\hat{\underline{x}}(k/k-1)$, $\underline{P}(k/k)$ and $\underline{P}(k/k-1)$ would be computed <u>and stored</u>. On the final iteration of the forward pass, with K = NN = 100,

$$\hat{\underline{x}}(100/100) = \hat{\underline{x}}(100/99) + \underline{G}(100)\, [\underline{z}(100) - \underline{H}\, \hat{\underline{x}}(100/99)]$$

i.e., we have computed and stored $\hat{\underline{x}}(100/100)$.

Now, the smoothing process commences in the reverse direction. Decrement k to k = NN-1 = 99, then

$$\hat{\underline{x}}(99/100) = \underbrace{\hat{\underline{x}}(99/99)}_{\text{stored}} + \underline{A}(99)\, [\underbrace{\hat{\underline{x}}(100/100)}_{\text{stored}} - \underbrace{\hat{\underline{x}}(100/99)}_{\text{stored}}]$$

and $$\underline{A}(99) = \underbrace{\underline{P}(99/99)}_{\text{stored}}\, \underline{\phi}^T\, \underbrace{\underline{P}(100/99)}_{\text{stored}}^{-1}$$

let k = NN-2 = 98, then

$$\underline{\hat{x}}(98/100) = \underbrace{\underline{\hat{x}}(98/98)}_{\text{stored}} + \underline{A}(98)\,[\underbrace{\underline{\hat{x}}(99/100)}_{\substack{\text{computed}\\ \text{last}\\ \text{iteration}}} - \underbrace{\underline{\hat{x}}(99/98)}_{\text{stored}}]$$

and $$\underline{A}(98) = \underbrace{\underline{P}(98/98)}_{\text{stored}}\ \underline{\emptyset}^{T}\ \underbrace{\underline{P}(99/98)}_{\text{stored}}{}^{-1}$$

Also, for each of the two preceding iterations,

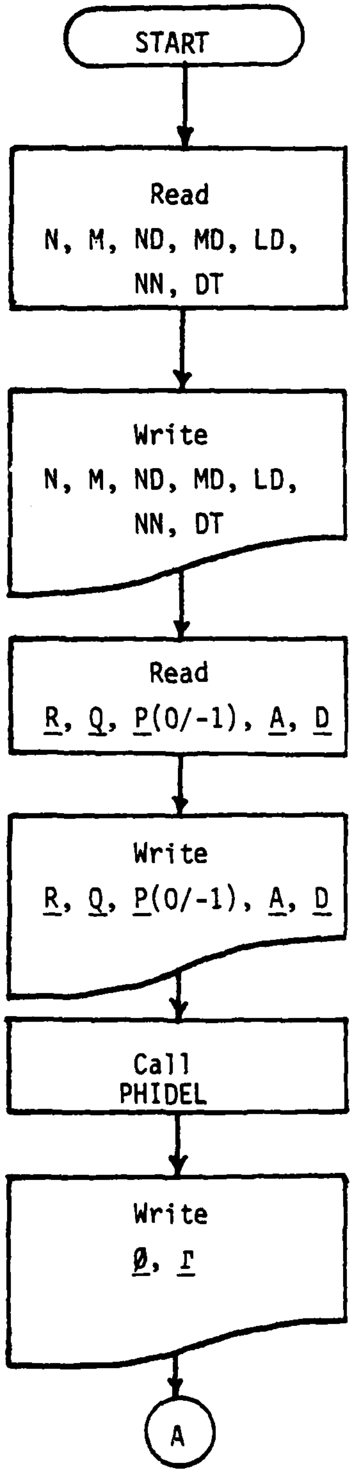
$$\underline{P}(99/100) = \underbrace{\underline{P}(99/99)}_{\text{stored}} + \underbrace{\underline{A}(99)}_{\text{computed}}\,[\underbrace{\underline{P}(100/100)}_{\text{stored}} - \underbrace{\underline{P}(100/99)}_{\text{stored}}]\ \underline{A}^{T}(99)$$

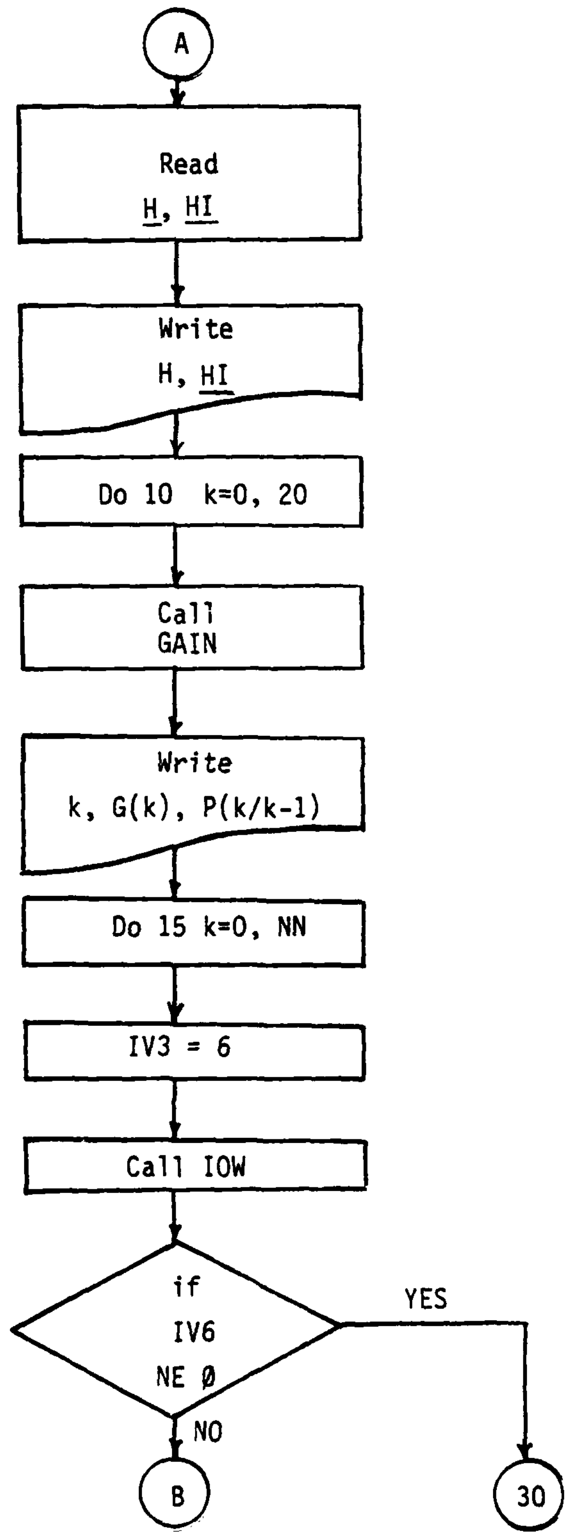
$$\underline{P}(98/100) = \underbrace{\underline{P}(98/98)}_{\text{stored}} + \underbrace{\underline{A}(98)}_{\text{computed}}\,[\underbrace{\underline{P}(99/100)}_{\text{computed}} - \underbrace{\underline{P}(99/98)}_{\text{stored}}]\ \underbrace{\underline{A}^{T}\,(98)}_{\text{computed}}$$

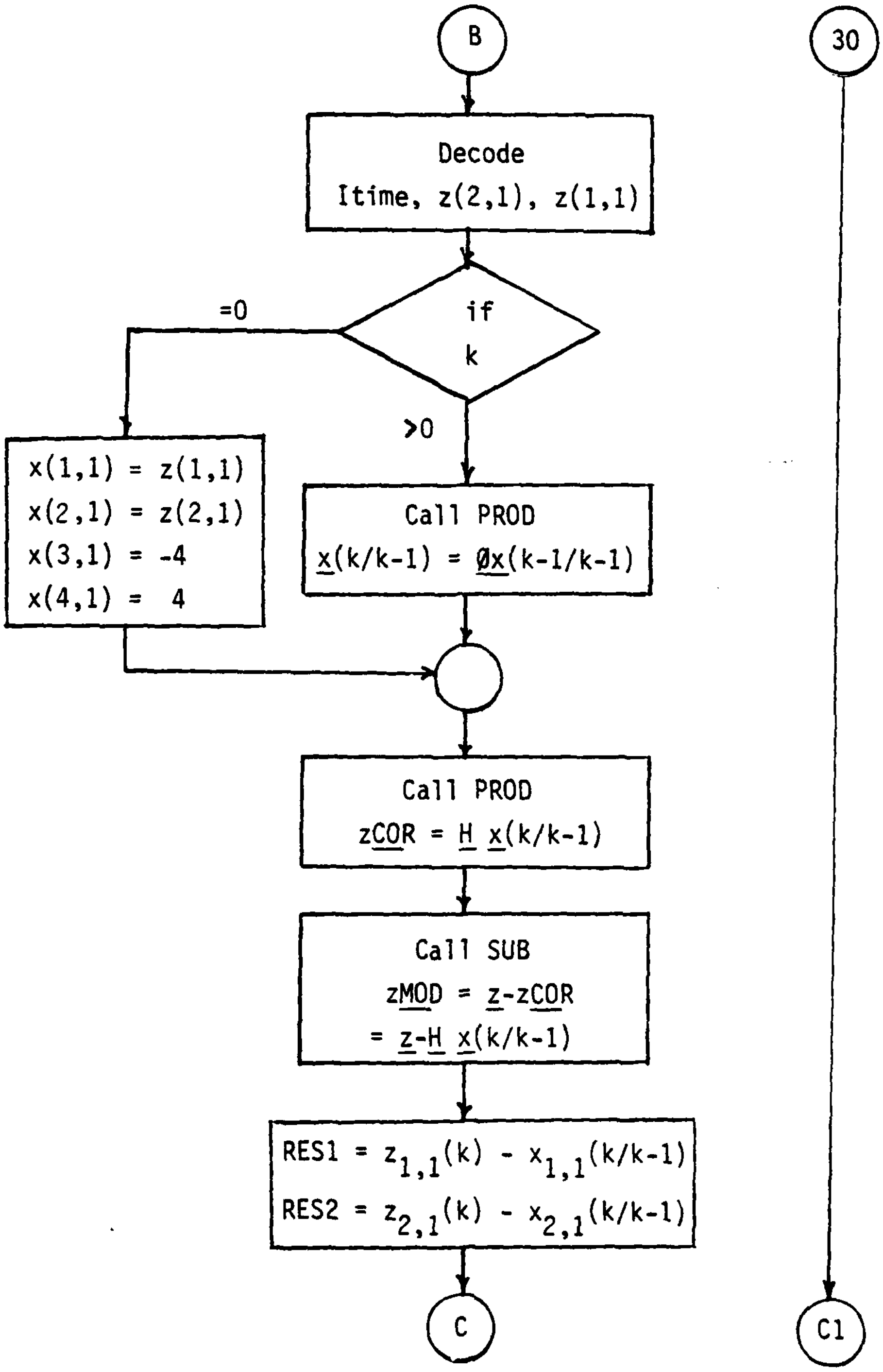
etc.

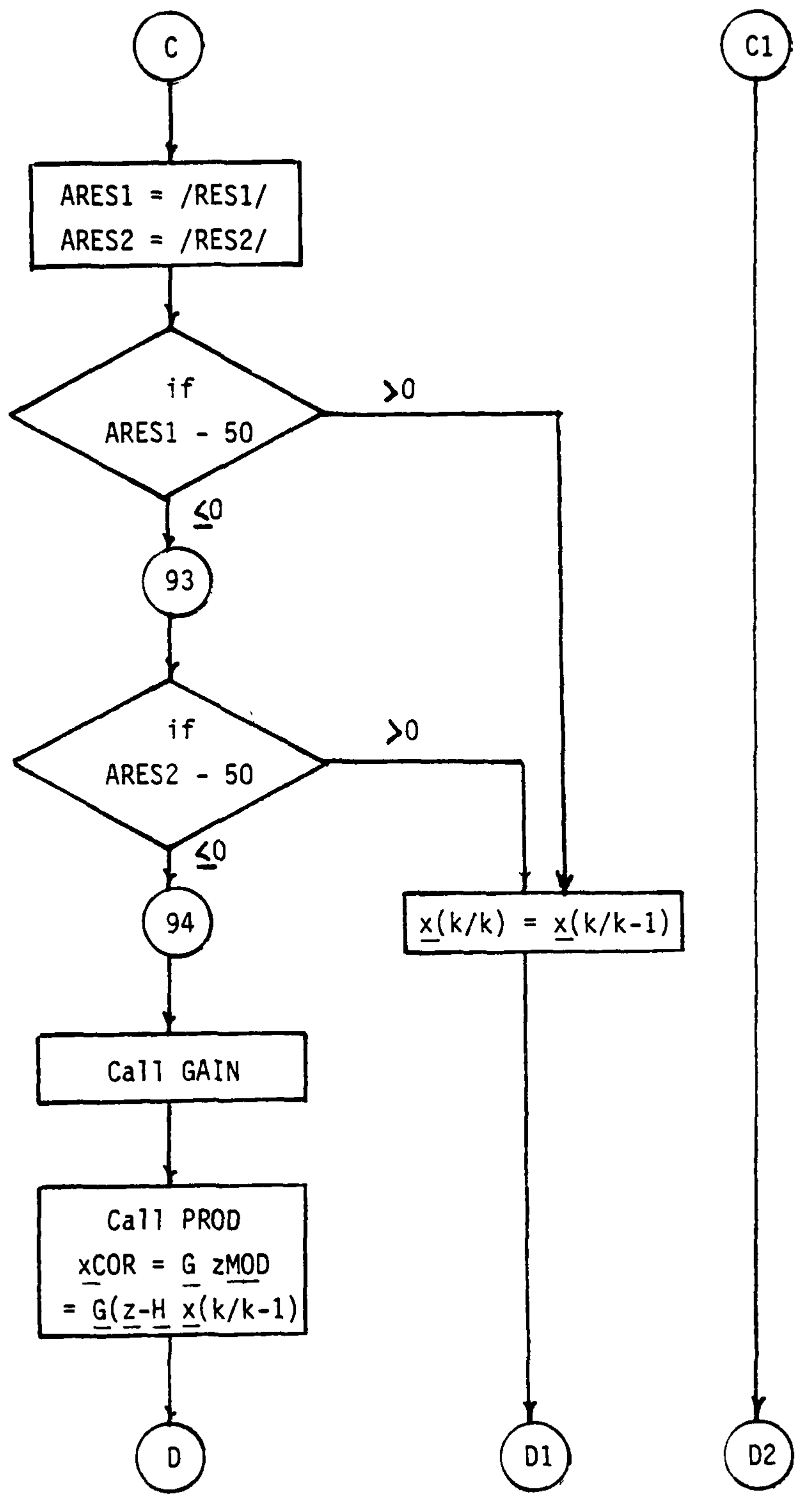
It is seen that the smoothing process does not involve the processing of actual measurement data. It does, however, utilize the complete filtering solution, and so fixed interval smoothing cannot be done real-time, on-line. It must be done after all the measurement data are collected. Consequently, computation speed will not be the most important factor. Storage requirements could, however, conceivably be, in that the quantities to be stored on the forward pass are arrays. It is seen that, should an exercise run in excess of 30 minutes, retention of the data at each mark could require in excess of 100K bytes of memory, which could limit the facilities upon which the processor could be utilized.

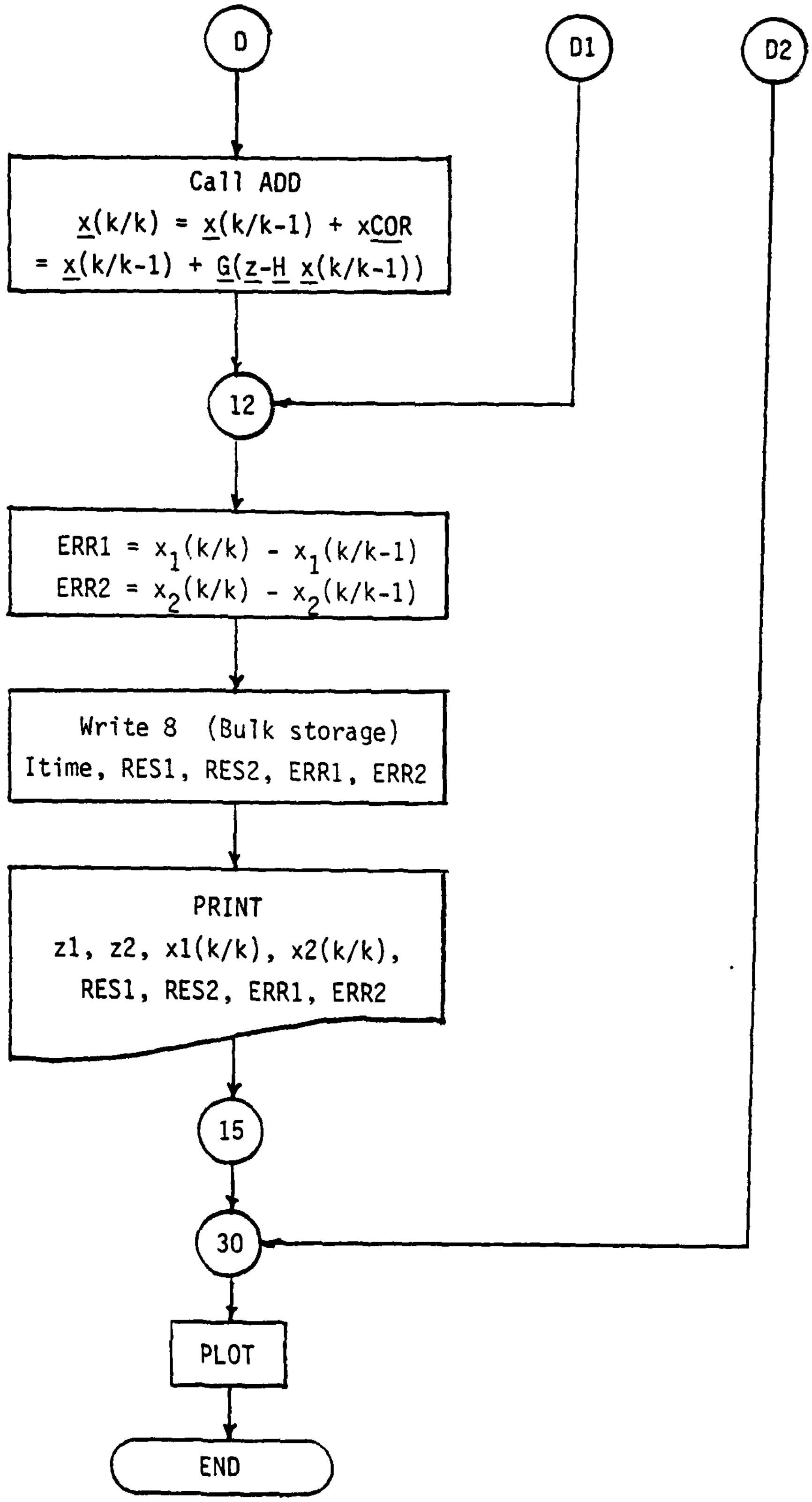
APPENDIX A: Processor Flowchart Main Program

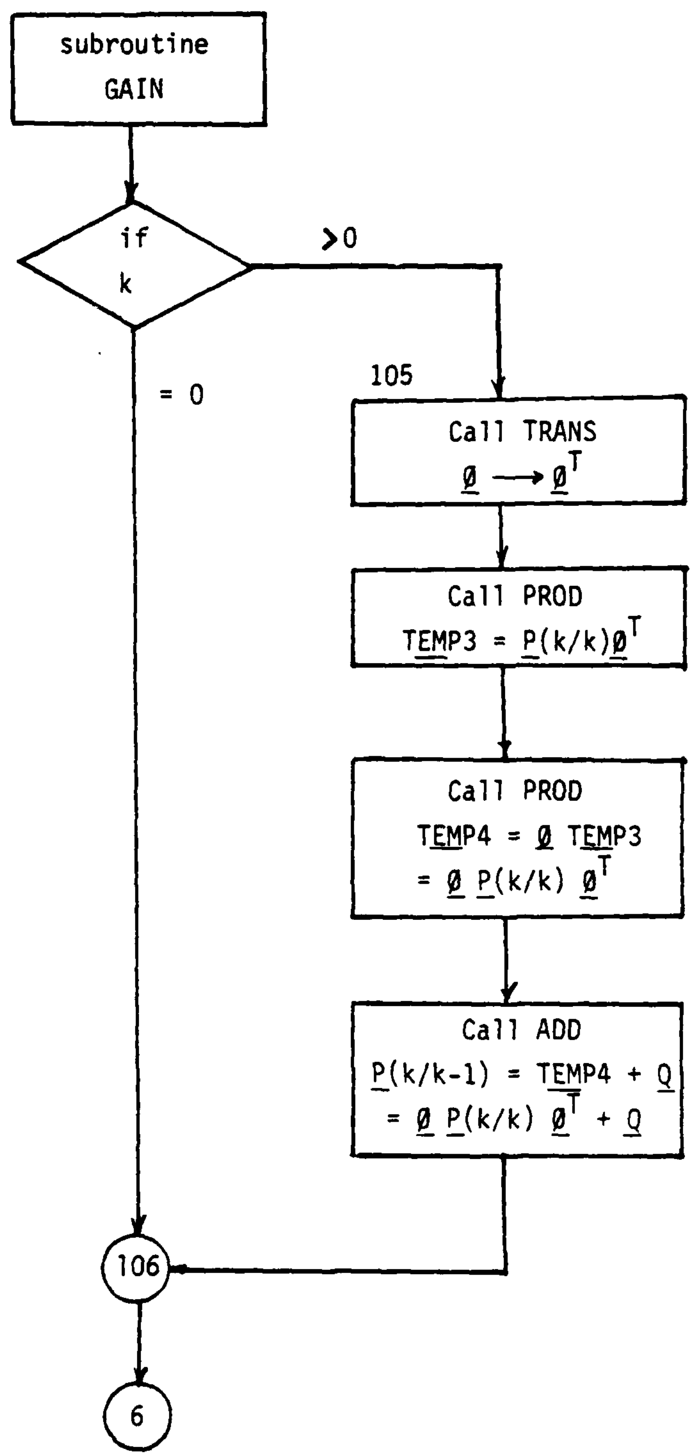


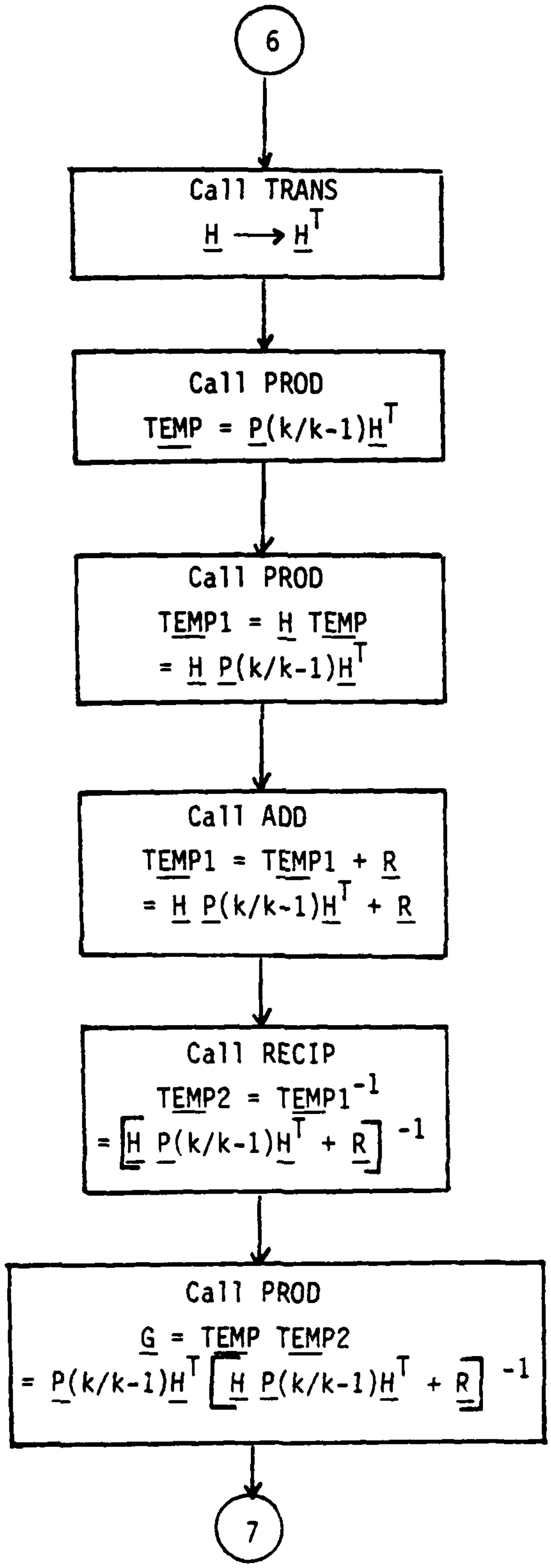

A
Read
H, HI
Write
H, HI
Do 10 k=0, 20
Call
GAIN
Write
k, G(k), P(k/k-1)
Do 15 k=0, NN
IV3 = 6
Call IOW
if
IV6
NE Ø
YES
NO
B
30

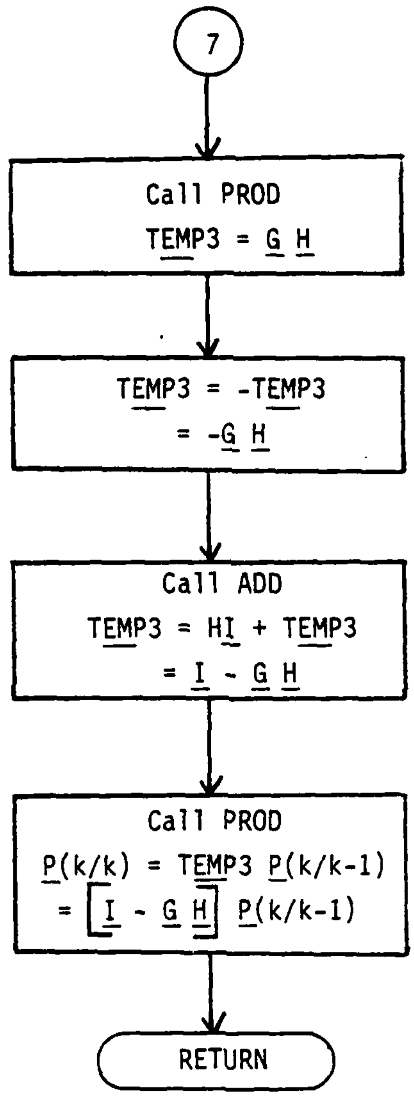

B
30
Decode
Itime, z(2,1), z(1,1)
if
k
=0
>0
x(1,1) = z(1,1)
x(2,1) = z(2,1)
x(3,1) = -4
x(4,1) = 4
Call PROD
x(k/k-1) = Øx(k-1/k-1)
Call PROD
zCOR = H x(k/k-1)
Call SUB
zMOD = z-zCOR
= z-H x(k/k-1)
RES1 = $z_{1,1}(k) - x_{1,1}(k/k-1)$
RES2 = $z_{2,1}(k) - x_{2,1}(k/k-1)$
C
C1


C
C1
ARES1 = /RES1/
ARES2 = /RES2/
if
ARES1 - 50
>0
≤0
93
if
ARES2 - 50
>0
≤0
94
x(k/k) = x(k/k-1)
Call GAIN
Call PROD
xCOR = G zMOD
= G(z-H x(k/k-1)
D
D1
D2


D
D1
D2
Call ADD
x(k/k) = x(k/k-1) + xCOR
= x(k/k-1) + G(z-H x(k/k-1))
12
ERR1 = x1(k/k) - x1(k/k-1)
ERR2 = x2(k/k) - x2(k/k-1)
Write 8 (Bulk storage)
Itime, RES1, RES2, ERR1, ERR2
PRINT
z1, z2, x1(k/k), x2(k/k),
RES1, RES2, ERR1, ERR2
15
30
PLOT
END


subroutine GAIN
if k
>0
= 0
105
Call TRANS
Ø → Ø^T
Call PROD
TEMP3 = P(k/k)Ø^T
Call PROD
TEMP4 = Ø TEMP3
= Ø P(k/k) Ø^T
Call ADD
P(k/k-1) = TEMP4 + Q
= Ø P(k/k) Ø^T + Q
106
6


6
Call TRANS
H → H^T
Call PROD
TEMP = P(k/k-1)H^T
Call PROD
TEMP1 = H TEMP
= H P(k/k-1)H^T
Call ADD
TEMP1 = TEMP1 + R
= H P(k/k-1)H^T + R
Call RECIP
TEMP2 = TEMP1^-1
= [H P(k/k-1)H^T + R]^-1
Call PROD
G = TEMP TEMP2
= P(k/k-1)H^T [H P(k/k-1)H^T + R]^-1
7


7
Call PROD
TEMP3 = G H
TEMP3 = -TEMP3
= -G H
Call ADD
TEMP3 = HI + TEMP3
= I - G H
Call PROD
P(k/k) = TEMP3 P(k/k-1)
= [I - G H] P(k/k-1)
RETURN

COMPUTER OUTPUT

Q = I, R = I

K	RAW R1	RAW R2	FILTERED R1	FILTERED R2
0	4622.4	4980.0	4622.4	4932.2
1	4623.3	4973.1	4627.4	4980.4
2	4633.0	4972.7	4632.1	4970.9
3	4636.0	4972.4	4635.3	4972.2
4	4639.7	4968.8	4639.6	4968.0
5	4639.3	4962.2	4640.5	4962.9
6	4646.7	4960.0	4646.0	4960.4
7	4643.3	4953.6	4643.1	4953.7
8	5650.7	4950.6	5474.2	4955.9
9	4636.0	4951.7	4378.1	4951.9
10	4660.3	4949.7	4681.8	4949.4
11	4657.7	4943.7	4653.4	4944.2
12	4666.5	4940.9	4643.3	4940.7
13	4673.6	4942.6	4660.7	4941.6
14	4674.3	4937.3	4669.7	4937.8
15	4676.3	4935.0	4673.7	4933.4
16	4601.0	4931.6	4614.6	4931.3
17	4604.6	4926.4	4601.0	4928.4
18	4607.9	4922.6	4602.9	4923.0
19	4611.3	4921.1	4603.5	4920.7
20	4616.1	4919.6	4614.4	4919.2
21	4702.0	4916.0	4690.3	4915.2
22	4702.7	4909.9	4700.7	4910.5
23	4709.3	4910.0	4714.4	4909.3
24	4711.9	4904.6	4715.3	4905.0
25	4711.6	4900.3	4710.9	4900.3
26	4718.1	4899.0	4718.0	4897.7
27	4724.1	4895.7	4722.7	4893.0
28	4725.7	4892.7	4726.2	4892.9
29	4728.5	4887.0	4728.7	4888.0
30	4733.2	4882.9	4733.0	4883.1
31	4736.7	4880.0	4776.7	4882.2
32	4742.4	4881.0	4742.0	4880.9
33	4746.5	4877.1	4745.6	4877.8
34	4748.5	4871.9	4749.0	4872.5
35	4753.7	4870.9	4750.5	4870.3
36	4757.9	4868.6	4757.8	4869.1
37	4752.8	4863.4	4762.5	4863.7
38	4763.4	4859.6	4754.0	4860.2
39	4767.7	4853.0	4767.5	4873.1
40	4772.0	4857.0	4772.3	4858.6
41	4773.2	4851.0	4773.7	4852.0
42	4700.0	4840.6	4714.3	4846.6
43	4703.3	4847.0	4700.3	4846.9
44	4706.9	4844.1	4701.3	4844.2
45	4712.1	4833.8	4703.3	4840.1
46	4713.9	4841.6	4714.3	4840.7
47	4718.3	4832.1	4718.0	4833.4
48	4802.4	4829.2	4727.9	4829.2
49	4804.6	4825.9	4303.7	4826.5
50	4807.9	4824.0	4310.0	4822.6

$\underline{Q} = \underline{I}, \ \underline{R} = \underline{I}$

RESIDUE 1	RESIDUE 2	ERROR 1	ERROR 2	TIME
.0	.0	.0	.0	105538
10.9	-7.1	9.0	-5.8	105539
5.0	-9.1	4.1	-7.5	105540
1.2	1.3	1.0	1.1	105541
.7	-1.1	.0	-.9	105542
-3.2	-4.1	-2.6	-3.3	105543
4.1	4.7	3.3	3.9	105544
-1.0	-.3	-.8	-.3	105545
999.8	-1.5	821.6	-1.2	105546
******	-1.0	******	-.3	105547
-118.0	1.4	-97.0	1.2	105548
124.9	-2.7	102.7	-2.3	105549
127.4	.8	104.7	.6	105550
72.3	5.2	59.5	4.7	105551
25.6	-2.9	21.0	-2.4	105552
8.3	-2.2	6.8	-1.8	105553
-76.6	1.7	-63.0	1.4	105554
20.0	-.1	16.4	-.1	105555
28.5	-3.1	23.4	-2.5	105556
18.6	2.2	15.3	1.8	105557
9.4	2.1	7.7	1.7	105558
85.4	-.9	70.2	-.7	105559
-22.3	-3.7	-18.3	-3.0	105600
-25.7	3.8	-21.1	3.1	105601
-20.5	-2.1	-16.9	-1.7	105602
-13.2	-1.1	-10.8	-.9	105603
.4	1.5	.3	1.2	105604
2.2	2.3	1.8	1.9	105605
-2.8	-1.2	-2.3	-1.0	105606
-1.3	-2.4	-1.1	-2.0	105607
1.4	-1.2	1.1	-1.0	105608
.1	4.3	.1	3.5	105609
2.0	1.6	1.7	1.3	105610
.0	-1.5	.0	-1.2	105611
-2.7	-3.3	-2.2	-2.7	105612
1.3	2.4	1.1	1.9	105613
.5	2.0	.4	1.6	105614
.6	-1.6	.5	-1.3	105615
-3.5	-3.3	-2.9	-2.7	105616
.7	2.5	.6	2.1	105617
1.8	2.1	1.5	1.7	105618
-3.1	-2.9	-2.5	-2.4	105619
-75.6	.1	-62.2	.0	105620
18.5	2.2	15.2	1.8	105621
28.0	-.3	23.0	-.2	105622
19.8	-1.7	16.2	-1.4	105623
8.5	4.9	7.0	4.0	105624
1.7	-7.3	1.4	-6.0	105625
81.4	.2	66.9	.2	105626
-20.7	2.1	-17.0	1.7	105627
-29.1	.9	-23.9	.8	105628

$\underline{Q} = 0.1\underline{I}, \quad \underline{R} = \underline{I}$

K	RAW R1	RAW R2	FILTERED R1	FILTERED R2
C	4622.4	4982.2	4622.4	4982.2
1	4629.3	4979.1	4624.7	4982.1
2	4633.C	4972.3	4628.8	4977.5
3	4636.C	4972.4	4633.1	4974.6
4	4639.7	4968.8	4637.6	497C.8
5	4639.9	4962.2	4640.1	4965.0
6	4646.7	4963.2	4645.C	4962.5
7	4648.9	4959.6	4648.7	4959.6
8	5652.3	4955.6	523C.4	4956.0
9	4656.6	4951.7	4986.8	4952.2
1C	4660.8	4949.7	4818.7	4949.3
11	4657.7	4943.7	4713.5	4944.7
12	4666.5	494C.8	4662.8	494C.9
13	4673.6	4942.6	4646.2	4940.3
14	4674.3	4937.3	4645.3	4937.5
15	4676.8	4933.C	4652.3	4933.7
16	46C1.0	4931.6	461E.5	4931.2
17	4604.6	4928.4	460C.2	4928.4
18	46C7.9	4922.5	4596.2	4923.8
19	4611.8	4921.1	4599.2	4920.8
2C	4616.1	4919.6	46C5.5	4918.7
21	47C2.0	4916.C	4660.0	4915.9
22	4702.7	49C9.8	4692.0	4911.2
23	47C9.8	4910.0	4711.8	49C9.0
24	4711.8	4904.6	472C.9	49C5.2
25	4711.6	49C0.3	4722.8	49C1.C
26	4718.1	4898.C	4725.C	4897.7
27	4724.1	4896.7	4728.0	4895.7
28	4725.7	4892.7	4729.4	4892.7
29	4728.5	4887.6	4730.8	4888.5
3C	4733.2	4882.9	4733.7	4883.8
31	4736.7	4883.C	4736.8	4881.7
32	4742.4	4881.2	4741.4	488C.1
33	4746.6	4877.5	4746.C	4877.4
34	4748.5	4871.9	4749.1	4873.1
35	4753.7	487C.9	4753.3	4870.5
36	4757.9	4869.5	4757.6	4868.6
37	4762.6	4865.4	4762.2	4865.7
38	4763.4	4859.6	4764.7	4861.C
39	4767.7	4858.5	4767.9	4858.1
4C	4772.6	4857.C	4772.1	4856.1
41	4773.2	4851.5	4774.3	4852.3
42	47CC.8	4848.6	4733.2	4848.8
43	4703.6	4847.3	4710.8	4846.6
44	47C6.9	4844.1	47C1.8	4843.9
45	4712.1	4839.9	47C2.0	4840.3
46	4715.8	4841.6	47C6.3	4839.8
47	4718.3	4832.1	4711.6	4834.4
48	48C2.4	4829.2	4763.8	483C.C
49	48C4.6	4826.9	4795.0	4826.6
5C	48C7.8	4824.C	4812.C	4823.6

$\underline{Q} = 0.1\underline{I}, \quad R = \underline{I}$

RESIDUE 1	RESIDUE 2	ERROR 1	ERROR 2	TIME
.0	.0	.0	.0	105538
10.9	-7.1	6.3	-4.1	105539
10.1	-12.3	5.8	-7.1	105540
6.9	-5.1	4.0	-3.0	105541
4.9	-4.7	2.8	-2.7	105542
-.5	-6.6	-.3	-3.8	105543
4.0	1.6	2.3	.9	105544
.4	.1	.2	.1	105545
1000.0	-.9	578.2	-.5	105546
-782.8	-1.1	-452.5	-.6	105547
-374.2	1.0	-216.3	.6	105548
-132.3	-2.4	-76.5	-1.4	105549
8.9	-.2	5.1	-.1	105550
64.9	5.5	37.5	3.2	105551
68.8	-.4	39.8	-.2	105552
58.1	-1.8	33.6	-1.0	105553
-36.6	.9	-21.2	.5	105554
10.3	.1	6.0	.0	105555
27.7	-3.0	16.0	-1.7	105556
30.0	.8	17.3	.5	105557
25.1	2.2	14.5	1.2	105558
99.6	.2	57.6	.1	105559
25.3	-3.3	14.6	-1.9	105600
-4.8	2.3	-2.8	1.3	105601
-21.6	-1.4	-12.5	-.8	105602
-26.5	-1.6	-15.3	-.9	105603
-16.4	.7	-9.5	.4	105604
-9.3	2.5	-5.4	1.4	105605
-8.8	.0	-5.1	.0	105606
-5.5	-2.1	-3.2	-1.2	105607
-1.1	-2.2	-.7	-1.3	105608
-.3	3.0	-.2	1.8	105609
2.4	2.7	1.4	1.6	105610
1.5	.1	.9	.1	105611
-1.5	-2.9	-.9	-1.7	105612
.9	1.0	.5	.6	105613
.7	2.1	.4	1.2	105614
1.0	-.6	.6	-.4	105615
-3.0	-3.3	-1.7	-1.9	105616
-.6	.9	-.3	.5	105617
1.2	2.1	.7	1.2	105618
-2.6	-1.8	-1.5	-1.0	105619
-76.7	-.5	-44.3	-.3	105620
-17.0	1.8	-9.8	1.0	105621
12.2	.5	7.0	.3	105622
23.9	-1.3	13.8	-.7	105623
22.4	4.3	13.0	2.5	105624
16.0	-5.5	9.2	-3.2	105625
91.6	-1.9	53.0	-1.1	105626
22.8	.6	13.2	.4	105627
-9.9	.9	-5.7	.5	105628

$\underline{Q} = 0.01\underline{I}, \quad \underline{R} = \underline{I}$

K	RAW R1	RAW R2	FILTERED R1	FILTERED R2
0	4622.4	4982.2	4622.4	4982.2
1	4629.3	4979.1	4622.4	4983.6
2	4633.0	4972.3	4624.3	4981.6
3	4636.0	4972.4	4627.4	4979.6
4	4639.7	4968.8	4631.3	4976.5
5	4639.9	4962.2	4634.5	4971.5
6	4646.7	4963.2	4639.5	4967.9
7	4648.9	4959.6	4644.0	4964.0
8	5652.3	4955.6	5017.2	4959.7
9	4656.6	4951.7	4936.2	4955.2
10	4660.8	4949.7	4864.3	4951.4
11	4657.7	4943.7	4801.7	4946.6
12	4666.5	4940.8	4753.9	4942.3
13	4673.6	4942.6	4719.4	4940.1
14	4674.3	4937.3	4694.3	4937.0
15	4676.8	4933.0	4677.8	4933.4
16	4601.0	4931.6	4639.3	4930.7
17	4604.6	4928.4	4613.3	4927.8
18	4607.9	4922.5	4597.4	4923.9
19	4611.8	4921.1	4589.6	4920.8
20	4616.1	4919.6	4588.1	4918.3
21	4702.0	4916.0	4621.0	4915.5
22	4702.7	4909.8	4648.5	4911.5
23	4709.8	4910.0	4672.7	4908.9
24	4711.8	4904.6	4691.8	4905.3
25	4711.6	4900.3	4705.3	4901.5
26	4718.1	4898.0	4716.7	4898.1
27	4724.1	4896.7	4726.2	4895.5
28	4725.7	4892.7	4732.7	4892.4
29	4728.5	4887.6	4737.2	4888.6
30	4733.2	4882.9	4741.1	4884.5
31	4736.7	4883.0	4744.3	4881.7
32	4742.4	4881.2	4747.7	4879.4
33	4746.6	4877.5	4751.1	4876.8
34	4748.5	4871.9	4753.5	4873.1
35	4753.7	4870.9	4756.6	4870.3
36	4757.9	4869.5	4759.8	4868.0
37	4762.6	4865.4	4763.4	4865.2
38	4763.4	4859.6	4766.0	4861.3
39	4767.7	4858.5	4768.9	4858.3
40	4772.6	4857.0	4772.5	4855.9
41	4773.2	4851.5	4775.0	4852.4
42	4700.9	4848.6	4749.8	4849.1
43	4703.6	4847.3	4730.9	4846.5
44	4706.9	4844.1	4718.1	4843.7
45	4712.1	4839.8	4711.0	4840.4
46	4715.8	4841.6	4703.0	4838.9
47	4718.3	4832.1	4707.7	4834.7
48	4802.4	4829.2	4739.3	4830.8
49	4804.0	4826.9	4765.1	4827.3
50	4807.8	4824.0	4785.7	4824.0

$\underline{Q} = 0.01\underline{I}$, $\underline{R} = \underline{I}$

RESIDUE 1	RESIDUE 2	ERROR 1	ERROR 2	TIME
.0	.0	.0	.0	105538
10.9	-7.1	4.0	-2.6	105539
13.7	-14.7	5.1	-5.4	105540
13.7	-11.5	5.1	-4.2	105541
13.3	-12.2	4.9	-4.5	105542
8.5	-14.7	3.1	-5.4	105543
11.4	-7.5	4.2	-2.8	105544
7.7	-7.0	2.8	-2.6	105545
1006.0	-6.5	370.9	-2.4	105546
-442.8	-5.5	-163.3	-2.0	105547
-322.4	-2.6	-118.9	-1.0	105548
-228.1	-4.6	-84.1	-1.7	105549
-138.5	-2.3	-51.1	-.9	105550
-72.6	4.0	-26.8	1.5	105551
-31.7	.5	-11.7	.2	105552
-1.5	-.7	-.6	-.3	105553
-60.6	1.5	-22.4	.6	105554
-13.8	1.0	-5.1	.4	105555
16.6	-2.2	6.1	-.8	105556
35.1	.5	12.9	.2	105557
44.4	2.1	16.4	.8	105558
128.3	.8	47.3	.3	105559
85.9	-2.6	31.7	-1.0	105600
58.7	1.8	21.6	.7	105601
31.8	-1.2	11.7	-.4	105602
10.0	-1.8	3.7	-.7	105603
2.2	-.1	.3	-.0	105604
-3.4	2.0	-1.3	.7	105605
-11.1	.4	-4.1	.2	105606
-13.8	-1.7	-5.1	-.6	105607
-12.6	-2.5	-4.6	-.9	105608
-12.0	2.0	-4.4	.8	105609
-8.5	2.8	-3.1	1.0	105610
-7.1	1.2	-2.0	.4	105611
-7.9	-1.8	-2.9	-.7	105612
-4.5	1.0	-1.7	.4	105613
-3.0	2.3	-1.1	.8	105614
-1.3	.2	-.5	.1	105615
-4.1	-2.8	-1.5	-1.0	105616
-2.0	.2	-.7	.1	105617
.1	1.7	.0	.6	105618
-2.9	-1.5	-1.1	-.5	105619
-77.5	-.8	-28.6	-.3	105620
-43.3	1.3	-16.0	.5	105621
-17.8	.6	-6.6	.2	105622
1.7	-.9	.6	-.3	105623
12.3	4.2	4.5	1.6	105624
16.9	-4.1	6.2	-1.5	105625
100.0	-2.5	36.9	-.9	105626
62.6	-.6	23.1	-.2	105627
35.1	-.0	12.9	-.0	105628

COMPUTER PROGRAM

```
      DIMENSION HI(4,4),Q(4,4),H(2,4),R(2,2),G(4,2),
     1PHI(4,4),PKK(4,4),PKKMI(4,4),EXKK(4,1),EXKKMI(4,1)
      DIMENSION DEL(4,2),A(4,4),D(4,2),D1(4,4),D2(4,4)
      DIMENSION ZCOR(2,1),ZMOD(2,1),XCOR(4,1)
      DIMENS'ON Z(2,1)
      DIMENSION GAMMA(4,2)
      DIMENSION IV1(2),IV7(6),BUFFER(2000)
      DIMENSION HEADER(4,3),DATA(4),YAXIS(4,2)
   80 FORMAT(5X,I6,15X,2F5.1)
      DATA IV1(1)/'7      '/IV1(2)/'       '/
      DATA ((HEADER(J,I),J=1,3),I=1,4)/6HRESIDU,6HE 1 VS,6H. TIME,
     16HRESIDU,6HE 2 VS,6H. TIME,6HERROR ,6H1 VS. ,6HTIME  ,
     26HERROR ,6H2 VS. ,6HTIME  /
      DATA ((YAXIS(J,I),J=1,2),I=1,4)/6H RESID,6HUE 1  ,6H RESID,
     16HUE 2  ,6H  ERRO,6HR 1   ,6H  ERRO,6HR 2   /
C
C
C     THIS PROGRAM COMPUTES THE FOLLOWING KALMAN FILTER GAIN AND COVARIANCE
C     EQUATIONS
C
C
C                                              -1
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT+R)
C
C     P(K/K) = (I-G(K)*H)*P(K/K-1)
C
C
C     P(K/K-1) = PHI*P(K-1/K-1)*PHIT+Q
C
C
C     AND UPDATES THE STATE ESTIMATES BY SOLVING
C
C
C     X(K/K) = X(K/K-1)+G(K)*(Z(K)-H*X(K/K-1))=EXKK,WHERE
C     X(1,1)=R1
```

```
C      X(2,1)=R2
C      X(3,1)=D(R1)/DT
C      X(4,1)=D(R2)/DT
C      Z(1,1) IS THE MEASURED (RAW) R1
C      Z(2,1) IS THE MEASURED (RAW) R2
C
C
C      X(K/K-1) = PHI(K/K-1)*X(K-1/K-1)+GAMMA(K/K-1)*W(K-1)=EXKKM1
C
C
C      Q(I,J) DEFINES  THE COVARIANCE OF THE PER SAMPLE RANDOM GAUSSIAN
C      EXCITATION OF THE PROCESS.
C
C      R(I,J) DEFINES THE RANDOM (GAUSSIAN) MEASUREMENT NOISE COVARIANCE
C      WHICH IS ADDED TO THE MEASURED SIGNALS.
C
C      H(I,J) IS THE IDENTITY MATRIX.
C
C      K IS THE DISCRETE POINT IN TIME AT WHICH THE STAGE OF THE PROCESS
C      IS BEING CONSIDERED.
C
C      PKK(I,J) = P(K/K) THE COVARIANCE OF EST ERROR AT TIME K, GIVEN K SAMPLES.
C
C
C      PKKM1(I,J) = P(K/K-1), THE COVARIANCE OF ESTIMATION ERROR AT TIME
C      K GIVEN K-1 SAMPLES.
C
C
C      N = NUMBER  STATES
C
C
C      M = NUMBER OF INPUTS
C
C
C      ND AND MD ARE DIMENSIONS OF READ-IN AND WRITTEN-OUT MATRICES.
```

```
C
C     NN = NUMBER OF ITERATIONS OF FILTER,  THIS WILL BE EQUAL TO THE NUMBER
C     OF DATA POINTS TO BE READ AND FILTERED, AND WILL CHANGE FROM JOB TO JOB.
C
C
      REWIND 8
      READ(5,50)N,M,ND,MD,LD,NN,DT
   50 FORMAT(6I5,F10.4)
      WRITE(6,7777)
 7777 FORMAT(1H1)
      WRITE(6,51)N,M,ND,MD,LD,NN,DT
   51 FORMAT(2X,2HN=,I5,5X,2HM=,I5,5X,3HND=,I5,5X,3HMD=,I5,5X,3HLD=,
     1I5,5X,3HNN=,I5,5X,3HDT=,F10.4)
      CALL MREAD(R,M,M,LD,LD)
      WRITE(6,53)
   53 FORMAT(//12H MATRIX R    /)
      CALL MWRITE(R,M,M,LD,LD)
      CALL MREAD(Q,N,N,ND,MD)
      WRITE(6,54)
   54 FORMAT(//12H MATRIX Q    /)
      CALL MWRITE(Q,N,N,ND,MD)
      CALL MREAD(PKKM1,N,N,ND,MD)
C
C
C     THIS IS THE INITIAL VALUE OF P(K/K-1), OR, P(0/-1) FOR K=0.
C
C
      WRITE(6,55)
   55 FORMAT(//13H MATRIX PKKM1/)
      CALL MWRITE(PKKM1,N,N,ND,MD)
      CALL MREAD(A,N,N,ND,MD)
      WRITE(6,65)
   65 FORMAT(//13H MATRIX A     /)
      CALL MWRITE(A,N,N,ND,MD)
      CALL MREAD(D,N,M,ND,LD)
      WRITE(6,70)
```

```
   70 FORMAT(//13H MATRIX D     /)
      CALL MWRITE(D,N,M,ND,LD)
      CALL PHIDEL(DT,N,M,A,D,PHI,DEL,D1,D2,ND,MD,LD)
      WRITE(6,58)
   58 FORMAT(//13H MATRIX PHI   /)
      CALL MWRITE(PHI,N,N,ND,MD)
      WRITE(6,62)
   62 FORMAT(//13H MATRIX DEL   /)
      CALL MWRITE(DEL,N,M,ND,LD)
      CALL CONST(1.0,DEL,N,M,GAMMA,ND,LD)
      WRITE(6,64)
   64 FORMAT(//13H MATRIX GAMMA/)
      CALL MWRITE(GAMMA,N,M,ND,LD)
      CALL MREAD(H,M,N,LD,MD)
      WRITE(6,59)
   59 FORMAT(//13H MATRIX H     /)
      CALL MWRITE(H,M,N,LD,MD)
      CALL MREAD(HI,N,N,ND,MD)
      WRITE(6,60)
   60 FORMAT(//13H MATRIX HI    /)
      CALL MWRITE(HI,N,N,ND,MD)
      WRITE(6,7777)
C
C
C     PRECOMPUTE THE GAIN SCHEDULE FOR PURPOSE OF PRINTING OUT, ONLY.
C
C
      DO 10 K=0,20
      CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HI,ND,MD,LD,K)
      L=K
      LM1=K-1
      WRITE(6,18)K
   18 FORMAT(//3H K=,I3)
   11 WRITE(6,99)
```

```
      99 FORMAT(//13H MATRIX G    /)
         CALL MWRITE(G,N,M,ND,LD)
         WRITE(6,21)L,LM1
      21 FORMAT(//3H P(,I3,1H/,I3,1H)/)
         CALL MWRITE(PKKM1,N,N,ND,MD)
      10 CONTINUE
C
C
C        COMMENCE THE MAIN ITERATION LOOP, K=0 INITIALIZES.
C        ALL RANGES ARE IN METERS. ALL RATES ARE IN METERS PER SECOND.
C
C
         DO 15 K=0,NN
         IV3=6
         CALL IOW(IV1,16,IV3,IV7,0,IV6)
         IF(IV6.NE.0)GO TO 30
         DECODE(36,80,IV7)ITIME,Z(2,1),Z(1,1)
         IF(K)2,1,2
C
C
C        INITIALIZE THE STATE ESTIMATE XEST(0/-1)=MEANX(0) ESTIMATE, WHICH
C        IN THIS CASE WILL BE THE FIRST MEASUREMENT FOR EXKKM1(1,1) AND (2,1),
C        AND INITIAL VELOCITIES FOR EXKKM1(3,1) AND (4,1).
C
C        FIRST MEASUREMENTS
C
C
       1 EXKKM1(1,1)=Z(1,1)
         EXKKM1(2,1) = Z(2,1)
C
C
C        INITIAL VELOCITIES
C
C
         EXKKM1(3,1)=-4.
         EXKKM1(4,1)=4.
```

```
      GO TO 3
C
C
C     ONE STEP PREDICTION XEST(K/K-1)=PHI*XEST(K-1/K-1)+GAMMA*W(K-1)
C
C
    2 CALL PROD(PHI,EXKK,N,N,1,EXKKM1,ND,MD,1)
C
C
C     UPDATE STATE ESTIMATE XEST(K/K)=XEST(K/K-1)+G(K)*(Z(K)-H*XEST(K/K-1))
C
C
    3 CALL PROD(H,EXKKM1,M,N,1,ZCOR,LD,ND,1)
      CALL SUB(Z,ZCOR,M,1,ZMOD,LD,1)
      RES1=Z(1,1)-EXKKM1(1,1)
      RES2=Z(2,1)-EXKKM1(2,1)
C
C
C     GATE RANGE MEASUREMENTS TO REDUCE IMPACT OF JITTER (IN MOST
C     SIGNIFICANT FIGURES) ON COVARIANCE OF MEASUREMENT NOISE. THIS
C     GATE WILL BE EFFECTIVE FOR SURFACE CRAFT ONLY, AND MUST BE EXPANDED
C     FOR HIGHER SPEED (AIRCRAFT) TRACKING.
C
C
C     ARES1=ABS(RES1)
C     ARES2=ABS(RES2)
C     IF(ARES1-50.)93,93,125
C  93 IF(ARES2-50.)94,94,125
C  94 CONTINUE
    4 CALL GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HI,ND,MD,LD,K)
      CALL PROD(G,ZMOD,N,M,1,XCOR,ND,LD,1)
      CALL ADD(EXKKM1,XCOR,N,1,EXKK,ND,1)
C     GO TO 12
C 125 EXKK=EXKKM1
C  12 CONTINUE
      ERR1=EXKK(1,1)-EXKKM1(1,1)
```

```
      ERR2 =EXKK(2,1)-EXKKM1(2,1)
      WRITE(8)K,ITIME,RES1,RES2,ERR1,ERR2
      IF(K)7,8,7
    8 WRITE(6,5)
    5 FORMAT(1H1,7X,1HK,5X,6HRAW R1,4X,6HRAW R2,4X,11HFILTERED R1,4X,
     *11HFILTERED R2,4X,9HRESIDUE 1,4X,9HRESIDUE 2,4X,7HERROR 1,4X,
     *7HERROR 2,4X,4HTIME)
    7 WRITE(6,6)K,Z(1,1),Z(2,1),EXKK(1,1),EXKK(2,1),RES1,RES2,ERR1,
     *ERR2,ITIME
    6 FORMAT(5X,I4,5X,F6.1,4X,F6.1,5X,F6.1,10X,F6.1,8X,F6.1,
     *4X,F6.1,8X,F6.1,4X,F6.1,6X,I6)
   15 CONTINUE
   30 CONTINUE
      ENDFILE 8
      REWIND 8
      CALL PLOTS (BUFFER,2000,9)
      CALL PLOT(1.5,-5.25,-3)
      DO 400 I=1,4
      REWIND 8
      IFLAG = 0
      CALL AXIS(0.,0.,6HTIME K,-6,8.,0.,0.,10.,10.)
      CALL AXIS(0.,-4.,YAXIS(1,I),12,8.,90.,-80.,20.,10.)
      CALL SYMBOL(0.25,4.25,0.5,HEADER(1,I),0.,18)
      DO 410 J=0,NN
      READ(8,END=430)K,ITIME,DATA
      X=K/10.
      Y=DATA(I)
      IF(Y.GT.80.)Y=80.
      IF(Y.LT.-80.)Y=-80.
      Y=Y/20.
      IF(K.EQ.80)GO TO 430
      IF(IFLAG.NE.0)GO TO 440
      IFLAG=1
      CALL PLOT (X,Y, 3)
  440 CALL PLOT (X,Y, 2)
```

```
  410 CONTINUE
  430 CALL PLOT (10.,0.,-3)
  400 CONTINUE
      CALL PLOT (10.,0.,999)
      REWIND 8
      WRITE(6,44)IV6
   44 FORMAT(13H1OW STATUS = ,I6)
      STOP
      END
```

```
      SUBROUTINE PHIDEL(T,N,M,A,B,PHI,DEL,D1,D2,ND,MD,LD)
      DIMENSION A(4,4),D(4,2),PHI(4,4),DEL(4,2),TERM(4,4),
     1COR(4,4),C(4,4),D1(4,4),D2(4,4),TEIL(4,4)
      TEST=1.E-7
      F=1.
      DO 10 IR=1,N
      DO 10 IC=1,N
      PHI(IR,IC)=0.
      PHI(IR,IR)=1.
      C(IR,IC)=A(IR,IC)
      TEIL(IR,IC)=T/2.D0*PHI(IR,IC)
   10 TERM(IR,IC)=T*PHI(IR,IC)
   50 DO 11 IR=1,N
      DO 11 IC=1,N
      COR(IR,IC)=T/F*C(IR,IC)
      PHI(IR,IC)=PHI(IR,IC)+COR(IR,IC)
      TEIL(IR,IC)=TEIL(IR,IC)+T/((F+1.   )*(F+2.   ))*COR(IR,IC)
   11 TERM(IR,IC)=TERM(IR,IC)+T/(F+1.   )*COR(IR,IC)
      DO 12 IR=1,N
      DO 12 IC=1,N
      C(IR,IC)=0.
      DO 12 K=1,N
   12 C(IR,IC)=C(IR,IC)+A(IR,K)*COR(K,IC)
      F=F+1.
      DO 13 IR=1,N
      DO 13 IC=1,N
      IF(ABS(COR(IR,IC)) .GT. TEST*ABS(PHI(IR,IC))) GO TO 50
   13 CONTINUE
      CALL PROD(TERM,D,N,N,M,DEL,ND,MD,LD)
      CALL PROD(TEIL,D,N,N,M,D2,ND,MD,LD)
      DO 14 IR=1,N
      DO 14 IC=1,M
   14 D1(IR,IC)=DEL(IR,IC)-D2(IR,IC)
      RETURN
      END
```

```
      SUBROUTINE GAIN(PKK,PKKM1,Q,R,PHI,H,N,M,G,HI,ND,MD,LD,K)
C
C
C     THIS SUBROUTINE COMPUTES THE OPTIMUM GAIN MATRIX AND THE ERROR
C     COVARIANCE
C
C
      DIMENSION PKK(4,4),Q(4,4),H(2,4),G(4,2),R(2,2),HI(4,4),HT(4,2),
     1TEMP(4,2),TEMP2(2,2),TEMP1(2,2),PHI(4,4),PHIT(4,4),PKKM1(4,4)
      DIMENSION TEMP3(4,4),TEMP4(4,4)
      IF(K) 106,106,105
  105 CONTINUE
C
C
C     NOTE HERE PKKM1(I,J) = P(K/K-1) WHERE
C     P(K/K-1)= PHI*P(K-1/K-1)*PHIT+Q
C
C
      CALL TRANS(PHI,N,N,PHIT,ND,MD)
      CALL PROD(PKK,PHIT,N,N,N,TEMP3,ND,MD,ND)
      CALL PROD(PHI,TEMP3,N,N,N,TEMP4,ND,MD,ND)
      CALL ADD(TEMP4,Q,N,N,PKKM1,ND,MD)
  106 CONTINUE
C
C
C                                                 -1
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)
C
C
      CALL TRANS(H,M,N,HT,LD,MD)
      CALL PROD(PKKM1,HT,N,N,M,TEMP,ND,MD,LD)
      CALL PROD(H,TEMP,M,N,M,TEMP1,LD,MD,LD)
      CALL ADD(TEMP1,R,M,M,TEMP1,LD,LD)
      CALL RECIP(M,0.0000001,TEMP1,TEMP2,KER,LD)
```

```
      IF (KER-2) 101,110,101
  110 WRITE(6,111)
  111 FORMAT (5HKER=2)
  101 CALL PROD(TEMP,TEMP2,N,M,M,G,ND,LD,LD)
C
C
C     NOTE HERE PKK(I,J) = P(K/K) WHERE
C     P(K/K) = (I-G(K)*H)*P(K/K-1)
C
C
      CALL PROD(G,H,N,M,N,TEMP3,ND,LD,ND)
      DO 108 I=1,N
      DO 108 J=1,N
  108 TEMP3(I,J)=-TEMP3(I,J)
      CALL ADD(HI,TEMP3,N,N,TEMP3,ND,MD)
      CALL PROD(TEMP3,PKKMI,N,N,N,PKK,ND,MD,ND)
      RETURN
      END

      SUBROUTINE ADD (A,B,N,M,C,ND,MD)
      DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
      DO 152 I=1,N
      DO 152 J=1,M
  152 C(I,J) = A(I,J) + B(I,J)
      RETURN
      END

      SUBROUTINE SUB (A,B,N,M,C,ND,MD)
      DIMENSION A(ND,MD),B(ND,MD),C(ND,MD)
      DO 152 I=1,N
      DO 152 J=1,M
  152 C(I,J) = A(I,J) - B(I,J)
      RETURN
      END
```

```
      SUBROUTINE PROD (A,B,N,M,L,C,ND,MD,LD)
      DIMENSION A(ND,MD),B(MD,LD),C(ND,LD)
      DO 1 I=1,ND
      DO 1 J=1,LD
    1 C(I,J)=0.
      DO 151 I=1,N
      DO 151 J=1,L
      DO 151 K = 1,M
  151 C(I,J) = C(I,J) + A(I,K)*B(K,J)
      RFTURN
      END

      SUBROUTINE TRANS(A,N,M,C,ND,MD)
      DIMENSION A(ND,MD),C(MD,ND)
      DO 153 I=1,N
      DO 153 J=1,M
  153 C(J,I) = A(I,J)
      RETURN
      END

      SUBROUTINE CONST(Q,A,N,M,C,ND,MD)
      DIMENSION A(ND,MD),C(ND,MD)
      IF(Q)11,10,11
   10 DO 100 I=1,N
      DO 100 J=1,M
  100 C(I,J) = 0.0
      RETURN
   11 IF(Q-1.0)13,12,13
   12 DO 120 I=1,N
      DO 120 J=1,M
  120 C(I,J) = A(I,J)
      RETURN
   13 IF(Q+1.0)15,14,15
   14 DO 140 I=1,N
      DO 140 J=1,M
```

```
140 C(I,J) = -A(I,J)
    RETURN
 15 DO 150 I=1,N
    DO 150 J=1,M
150 C(I,J) = Q*A(I,J)
    RETURN
    END

    SUBROUTINE RECIP(N,EP,B,X,KER,M)
    DIMENSION A(2,2),X(M,M),B(M,M)
    CALL CONST(1.,B,N,N,A,2,2)
    DO 1 J=1,M
    DO 1 I=1,M
  1 X(I,J)=0.
    DO 2 K=1,N
  2 X(K,K)=1.
 10 DO 34 L=1,N
    KP=0
    Z=0.
    DO 12 K=L,N
    IF(Z.GE.ABS(A(K,L))) GO TO 12
 11 Z=ABS(A(K,L))
    KP=K
 12 CONTINUE
    IF(L.GE.KP) GO TO 20
 13 DO 14 J=L,N
    Z=A(L,J)
    A(L,J)=A(KP,J)
 14 A(KP,J)=Z
    DO 15 J=1,N
    Z=X(L,J)
    X(L,J)=X(KP,J)
 15 X(KP,J)=Z
 20 IF(ABS(A(L,L)).LE.EP) GO TO 50
 30 IF(L.GE.N) GO TO 34
 31 LP1=L+1
```

```
      DO 36 K=LP1,N
      IF(A(K,L).EQ.0.) GO TO 36
   32 RATIO = A(K,L)/A(L,L)
      DO 33 J=LP1,N
   33 A(K,J)=A(K,J)-RATIO*A(L,J)
      DO 35 J=1,N
   35 X(K,J)=X(K,J)-RATIO*X(L,J)
   36 CONTINUE
   34 CONTINUE
   40 DO 43 I=1,N
      II=N+1-I
      DO 43 J=1,N
      S=0.
      IF(II.GE.N) GO TO 43
   41 IIP1=II+1
      DO 42 K=IIP1,N
   42 S=S+A(II,K)*X(K,J)
   43 X(II,J)=(X(II,J)-S)/A(II,II)
      KER=1
      RETURN
   50 KER=2
      RETURN
      END
      SUBROUTINE MREAD(A,N,M,ND,MD)
      DIMENSION A(ND,MD)
      DO 10 I=1,N
   10 READ(5,20)(A(I,J),J=1,M)
   20 FORMAT(8F10.5)
      RETURN
      END
      SUBROUTINE MWRITE(A,N,M,ND,MD)
      DIMENSION A(ND,MD)
      DO 10 I=1,N
   10 WRITE(6,20)(I,J,A(I,J),J=1,M)
   20 FORMAT(6(3X,1H(,I2,1H,,I2,2H)=,1PE10.3))
      RETURN
      END
```

LIST OF REFERENCES

1. Department of the Navy, Naval Torpedo Station, Keyport, Washington, Report 1148, DM-40 Autotape Evaluation, October, 1971.

2. Kirk, D.E., Optimal Estimation: An Introduction to the Theory and Applications, Naval Postgraduate School, 1975.

3. Gelb, A., Applied Optimal Estimation, M.I.T. Press, 1974

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor H.A. Titus, Code 52Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	18
5.	LCDR Benjamin E. Julian Puget Sound Naval Shipyard Bremerton, Washington 98314	1
6.	Commanding Officer Naval Torpedo Station Keyport, Washington 98345 ATTN: Code 70	2